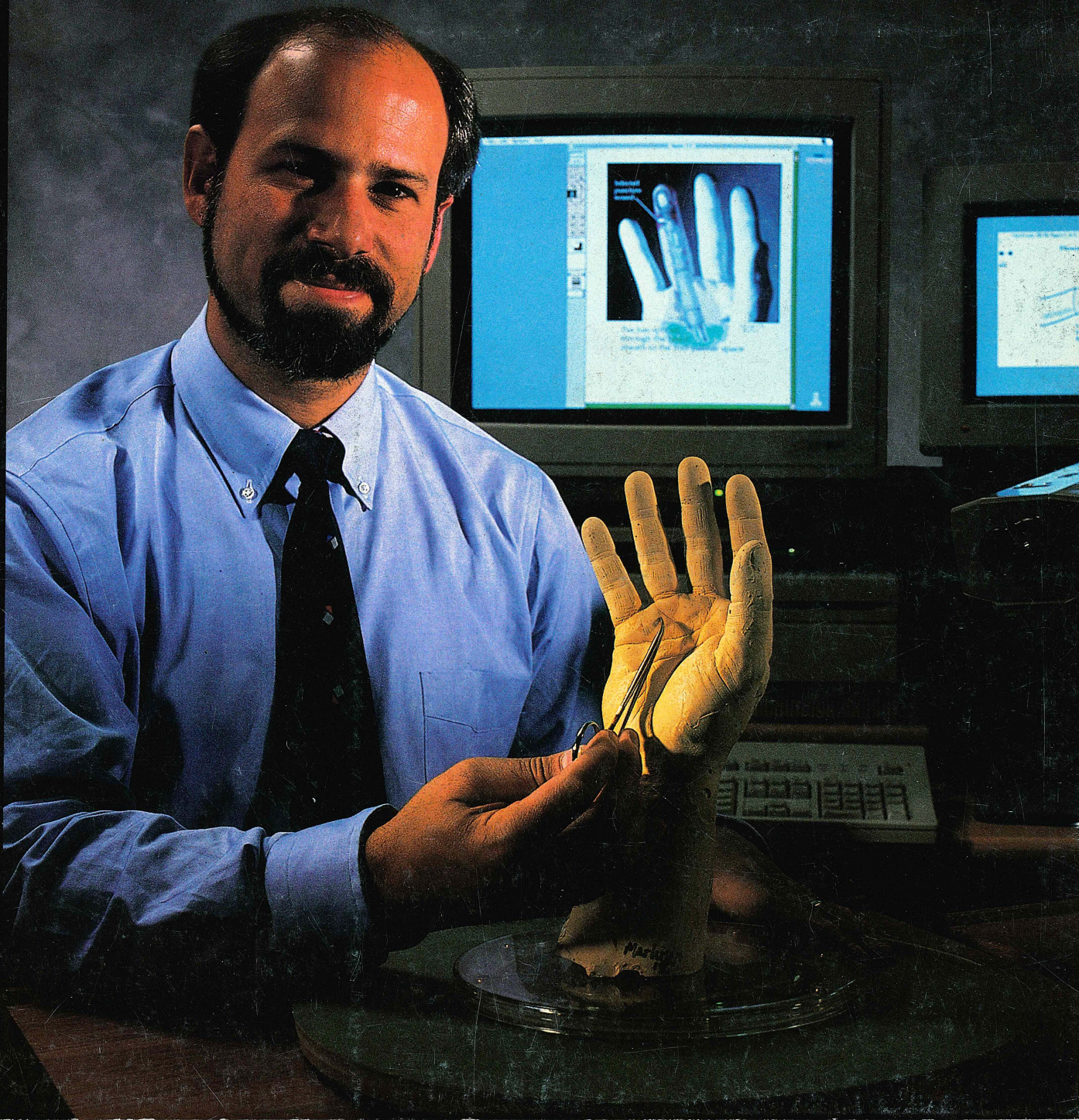
1988
Volume 4 Number 2

WHEELS FOR THE MIND

An Apple Higher Education Journal Prepared at Boston College



Save 25% off the single-copy price with this special introductory offer!

Just mail the attached card or call:

800-354-8400

(In California, call 800-331-4164)

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 453 CUPERTINO, CA

APPLE COMPUTER, INC.
WHEELS FOR THE MIND
P.O. BOX 1834
ESCONDIDO, CA 92025

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 453 CUPERTINO, CA

APPLE COMPUTER, INC.
WHEELS FOR THE MIND
P.O. BOX 1834
ESCONDIDO, CA 92025

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 453 CUPERTINO, CA

APPLE COMPUTER, INC.
WHEELS FOR THE MIND
P.O. BOX 1834
ESCONDIDO, CA 92025

Save 25% with this introductory offer.

☐ YES! I want to save 25% and more over the single-copy price by ordering a one or two-year subscription to *Wheels for the Mind.*

Please enter a subscription for the following: ☐ 1 year for $12 ☐ 2 years for $21

I understand that my subscription will start as soon as payment is received. (Please allow 6 to 8 weeks to receive the first issue.)

Name _____

Institution/Company _____

Address _____

City _____ State _____ Zip _____

☐ Check or money order enclosed (Make check payable to Apple Computer, Inc.)

☐ Credit Card Payment Visa

Card Number and Expiration Date MasterCard _____

NOTE: Orders from outside the United States must be accompanied by $24 in U.S. funds for a one-year subscription, which includes first-class postage and handling.

X _____

Signature Date

3089

Save 25% with this introductory offer.

☐ YES! I want to save 25% and more over the single-copy price by ordering a one or two-year subscription to *Wheels for the Mind.*

Please enter a subscription for the following: ☐ 1 year for $12 ☐ 2 years for $21

I understand that my subscription will start as soon as payment is received. (Please allow 6 to 8 weeks to receive the first issue.)

Name _____

Institution/Company _____

Address _____

City _____ State _____ Zip _____

☐ Check or money order enclosed (Make check payable to Apple Computer, Inc.)

☐ Credit Card Payment Visa

Card Number and Expiration Date MasterCard _____

NOTE: Orders from outside the United States must be accompanied by $24 in U.S. funds for a one-year subscription, which includes first-class postage and handling.

X _____

Signature Date

3089

Save 25% with this introductory offer.

☐ YES! I want to save 25% and more over the single-copy price by ordering a one or two-year subscription to *Wheels for the Mind.*

Please enter a subscription for the following: ☐ 1 year for $12 ☐ 2 years for $21

I understand that my subscription will start as soon as payment is received. (Please allow 6 to 8 weeks to receive the first issue.)

Name _____

Institution/Company _____

Address _____

City _____ State _____ Zip _____

☐ Check or money order enclosed (Make check payable to Apple Computer, Inc.)

☐ Credit Card Payment Visa

Card Number and Expiration Date MasterCard _____

NOTE: Orders from outside the United States must be accompanied by $24 in U.S. funds for a one-year subscription, which includes first-class postage and handling.

X _____

Signature Date

3089

We Want to Hear!

We're looking for a good story. Apple is always looking to hear from people who are doing exciting things with our computers. We want to hear from you, talk with you, and be inspired by your work. The information we gather can then be used in *Wheels for the Mind,* posted on AppleLink for other Macintosh users to read, included as success stories in the *AppleGram* mailings, and otherwise used as reference material. Who knows? We may even call you and invite you to show your work at one of our meetings, conferences, or trade shows, or to appear in an Apple brochure.

We are interested in information regarding software development, networking and communications, labs, multimedia, authoring, and productivity solutions, in such disciplines as English, computer science, business, medicine, humanities, foreign languages, and music. We want to hear about anything you think would be of interest to your colleagues at other institutions or to other Macintosh users. You can include any handouts, documentation, illustrations, screen dumps, or other materials that you feel are important to understanding your work. Please provide us with a comprehensive description of your project, including the following information:

- How did your project get started? (history)
- What hardware and software configurations are you using? (include any third-party equipment)
- What resources are required to complete your project?
- How did you achieve funding for your project?
- What are the expected/desired results of your project?
- How has computing enhanced or altered the learning process, your department, the school environment, or other areas?
- How can your experiences benefit other schools or Macintosh users?

Please fill in the information below and attach this sheet to the front of your report. And don't be surprised if an Apple person calls you sometime to hear more about your project.

Name _____

Address _____

City _____ State _____ Zip _____

Institution _____ Dept. _____ Title _____

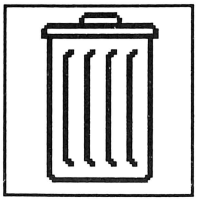
Telephone _____ Other _____

I would be willing to speak or make a presentation at a conference session_____

Please return to: Apple Computer, Inc., Wheels for the Mind
20525 Mariani Ave., M/S 36L, Cupertino, CA 95014
Attn: Monica Sabo, AppleLink: SABO1

WHEELS FOR THE MIND

Editorial



Hello, everyone. This issue of *Wheels for the Mind* will explain some of the reasons behind the excitement currently brewing in the academic environment. The long-held dream of providing innovative approaches to learning through multimedia presentations is now becoming a reality. And, as you will see in this issue, Apple Computer has had an advanced technology group exploring the possibilities of this area for quite some time. The team's observations, speculations, plans, and dreams make for interesting reading.

Our next issue will be a very special one, focusing entirely on HyperCard. It will come out in August to commemorate the first anniversary of this great tool. We expect to have some inside scoops about what's planned for HyperCard in the following year. In addition, we will cover a wide variety of stacks that have been created at colleges and universities and by third-party vendors, as well as providing you with information on how to obtain some public domain and shareware stacks. The issue will be both exciting and unique. Be watching for it!

In the fall, we will be introducing a new look to *Wheels for the Mind.* Based on feedback from many of our readers, we will be organizing the issue into three distinct sections: Instruction, Research, and Administration. Each section will showcase work going on at various campuses, and offer tips, information on new products, project reports, and resources. We also have a few surprises planned.

As always, your suggestions, contributions, and project reports are welcome. *Wheels* is *your* publication. By sharing your work and ideas with others, we all grow much wiser. Please send along any material to:

Peter Olivieri
Wheels for the Mind
Boston College F430
Chestnut Hill, MA 02167
AppleLink: A0130
BITNET: olivieri@bcvms

Wheels for the Mind

Editor:	Peter Olivieri
Copy editor:	Anne Kaplan-Neher
Contributors:	Anne Kaplan-Neher
	Chris Kim
	Monica Sabo
	The higher education community
Production:	Cynthia Berglund
	Laura Kurchin
Coordinators:	Mary Pat Davey
	Diane Tapscott
Apple contacts:	Lisa DiDomenico
	Monica Sabo

Wheels for the Mind is prepared on a Macintosh computer using Aldus PageMaker software, and printed out on an Apple LaserWriter printer.

An Apple Higher Education journal produced in collaboration with Boston College.
Published quarterly.

© 1988 Apple Computer, Inc.
Apple, the Apple logo, AppleLink, AppleShare, AppleTalk, Apple IIGS, A/UX, HyperCard, ImageWriter, LaserWriter, Macintosh, and Wheels for the Mind are registered trademarks of Apple Computer, Inc. APDA, AppleCD SC, Apple DeskTop Bus, Aristotle, Finder, HyperTalk, Inter•Poll, LaserShare, LocalTalk, MacAPPC, MacWorkstation, MultiFinder, ProDOS, Stackware, and Switcher are trademarks of Apple Computer, Inc. MacAdemia, Macworld, and the Wheels for the Mind logo are service marks of Apple Computer, Inc. NuBus is a trademark of Texas Instruments. PageMaker is a registered trademark of Aldus Corporation. PostScript is a registered trademark of Adobe Systems, Incorporated. UNIX is a registered trademark of AT&T Information Systems, Inc.

Mention in *Wheels for the Mind* does not constitute an endorsement by Apple Computer, Inc. Pricing and availability of third-party products is given according to information received and is subject to change.

Table of Contents

Volume 4, Number 2

Focus **1**

Learning Tomorrow: Multimedia in Education 1
Current Multimedia Projects 4
An Introduction to Project Jefferson 4
Brown University Intermedia Project 13
New York University Computer-Interactive-Video Project 15
Hypertext and "Intertextuality" at the College of Wooster 17
Macintosh and Medical Instruction at UCSF: A Profile of Bob Markison 21
Multimedia Feedback: Share Your Hopes and Concerns 27
Multimedia Resources 29
Perspectives on Multimedia and the Future of Computers in Education 30

Spotlight **37**

University of Nebraska: AppleShare and the College of Business 37
Drexel's Co-op Disk: Preparing for a Job Electronically 39
Drexel's Publication Development Center 40
Microcomputers in Education: A Special Place for HyperCard 41
University of Wyoming: Demonstration of Fundemental Kinematic Systems 42
Hamilton College: A Project to Improve the Reading/Communication Process 43
San Jose State University: Spartan Bookstore Macintosh Training Lab 46
Oregon State University: Speedcode Universe 47
Ohio State: An Engineering Approach to Software Design 50
California State University: Macintosh and Student Organizations 53

Table of Contents

Notre Dame: Computing Center Macintosh Lab 54
Gonzaga University: Attention Schools of Education 56
Tulane University: Macintosh Computers and Witchcraft 58
Astrophysical Research Consortium (ARC): Remote Astronomy and the Macintosh 59
California State University: Advanced Technology Laboratory 64
Tarrant County Junior College: Computerized Freshman Composition 66
Texas Christian University: Vertebrate Paleontology, Geology—and Hacking 69

News and Notes **73**
Joint Development Effort with DEC 73
Apple's Perspective on the DEC Agreement 73
Apple Acquires Network Innovations 75
University of Illinois Wins Apple's Project 2000 Contest 75
Desktop Publishing Conference 77
Resources in Special Education and Rehabilitation 78
EDUCOM Brochure on Software Use 78
1988 ACM Scholastic Programming Contest 79
Philadelphia Is Nuts About Hypermedia 81
Attention Language Teachers 81
Adding Memory to Your Macintosh 82
Easy Access to the Macintosh 83
Help Sought for Decision Support System 84
ICEC/Apple Fellowship Program 84

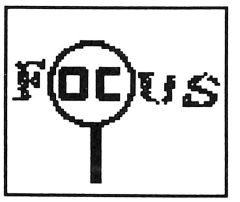
Products **91**
New Products from Claris 91
New Products from Apple 94
Keeping Track of Student Grades 102
New Bibliographic Tools 102
LaserCount 103

HyperCard **105**
HyperCard Scripts and Tips 105
HyperCard Stacks 122

Resources **135**

Project Reports **143**

Focus



This issue, the **Focus** section takes a look at multimedia—a unique blend of diverse technologies that adds up to exciting new potential for instructors, researchers, and students. Reflecting the breadth and variety of the topic, the articles are from a real cross section of the community involved in multimedia experimentation. We hope that you find their views and ideas stimulating.

Learning Tomorrow: Multimedia in Education

Sueann Ambron is the manager of New Technology in Education at Apple Computer. Her responsibilities include investigating new technologies for the education market and building prototype products that demonstrate the promise of new technologies. The following comments are taken from the introduction of Interactive Multimedia: Visions for Developers, Educators, and Information Providers (*Microsoft Press, 1988*), *which she coedited with Apple's Kristina Hooper.*

Multimedia is a powerful tool; it will change the way we look at knowledge and give us a new vision of reality. The changes are already happening. They started with the wide acceptance of the microcomputer and continue as we move toward videocomputers, interactive television, and electronic books. Three industries—computers, television, and publishing—that were quite separate during the sixties started to overlap in the seventies and eighties. By the nineties, the three industries will be even more intertwined. The products of these industries provide the content and media for education. Consequently, the new reality of multimedia will change education.

Multimedia is a powerful tool; it will change the way we look at knowledge and give us a new vision of reality.

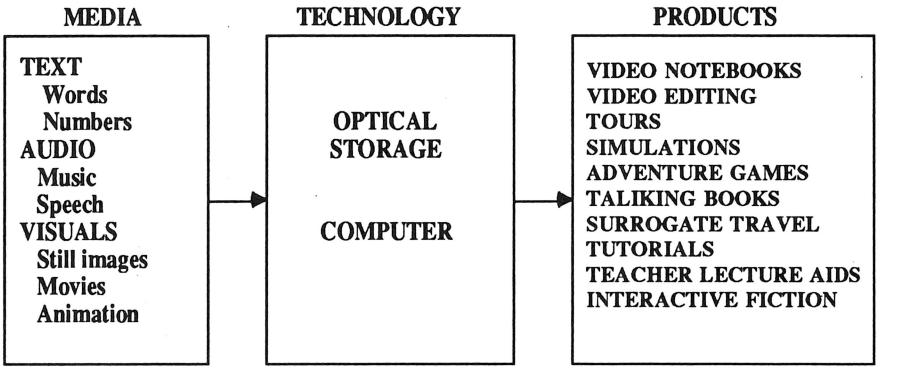
Multimedia can be examined in three parts: media, technology, and products.

Multimedia will provide instructors, students, and others with a dramatic new environment for presentations. They will no longer be bound by the limits of illustrating processes on a chalkboard or in slide collections. Vast libraries of audio, visual, and text material will be easily accessed by the computer. In addition, the user can integrate such existing computer programs as word processors, graphics, and data files. Instructors and students will have the opportunity to create their own classroom videos for presentation

and personal video notebooks. Students and teachers alike will be able to become "video producers" or "video publishers," utilizing the vast databases at their command for images, sounds, and text.

Researchers will use multimedia for new approaches to scholarship

Researchers will use multimedia for new approaches to scholarship. Multimedia will allow a level of analysis and visualization not possible with traditional film, audio, or print materials. Multiple scholars will be able to work on the same multimedia database simultaneously, combining their analysis of different aspects of the field. The business world will discover that "desktop video" is a powerful method of communicating within the company and to customers.



Multimedia Elements

It is a powerful idea, one whose time has arrived. It will significantly alter the way we teach and learn in the future.

Recently, we had the opportunity to speak with Sueann about multimedia in education.

Q. What is Apple's position/strategy toward multimedia?

Sueann: At Apple, there are several groups working on different aspects of multimedia technology and applications. Kristina Hooper and I head a group called the Multimedia Group. Basically, we see it as resembling a user group of the future—we're working with technologies that will become part of, or peripherals to, personal computers. We build working multimedia prototypes.

Our group collaborates with a broad cross section of people involved in multimedia.

One thing that's important to understanding this work is that we are design driven, or application driven—not driven by the technology. Our prototypes suggest key software tools, interfaces, and new hardware. We collaborate with a cross section of people involved in multimedia, including computer scientists, video engineers, and educators, as well as filmmakers and TV producers. Many of the prototypes that we develop in the Multimedia Group will eventually be products of the collaborating companies.

Q. What can multimedia do for educators?

Sueann: There are several ways of thinking about this. Basically, we're talking about a new kind of computer experience. Multimedia allows access to a wide variety of information types. The technology is available. How effective it proves to be is now a function of design; it's not a technological barrier, it's a design barrier. At this point, much of what's out there is primitive, but it will get better. And even now, some of it's pretty good. For example, a setup with HyperCard® and a videodisc lets you manage audiovisual materials in a lecture setting.

One thing that I find exciting is that, as the multimedia materials develop, the distinctions between teacher and student or expert and novice will blur; everyone will have access to vast stores of information, and be able to make their own way through these databases. The old standard of the "teacher as repository of knowledge" will be replaced by the "teacher as expert guide." A related development will be the mutability of the knowledge base; data will not be static but open-ended—both teachers and students will be able to add to it.

Finally, to me, the most compelling aspect is the ready access to full-motion video and sound. This will allow instructors to vastly increase the emotional and motivational appeal of their material, which, in turn, should have a dramatic effect on the impact that it has on students and their eventual information retention.

To me, the most compelling aspect is the ready access to full-motion video and sound.

Q. How can teachers best prepare themselves for use of multimedia in instruction?

Sueann: They should just start—just do it. Expose yourself to the technology and just leap in and play around with it. In terms of the actual equipment, the best place to begin is probably with a videodisc and/or CD-ROM and HyperCard.

Q. What are some of the multimedia projects that excite you most?

Sueann: The work we're doing. Seriously. The things that we're doing in the Multimedia Group are probably the most exciting things I've seen in years of involvement with technology. For example, there's the Grapevine project. It was started by a San Francisco high school teacher and a librarian who together dreamed of enhancing the study of John Steinbeck's *The Grapes of Wrath* by collecting and analyzing material in every medium—books, films, television documentaries, still photographs, record albums and sound tapes, filmstrips, magazine and newspaper articles, and more. That dream is now being realized, and the implications for enhancing instruction are incredibly exciting.

I'm not the only one who feels this kind of excitement about the Multimedia Group. We have a number of visitors, and we can't get them out once they're in—they just want to stay. We've had people from industry—from Pioneer, for example—and people from universities such as Harvard; it's been, again, a broad cross

section. And all of these people are just as tremendously excited by what's possible—and by what we're doing—as we are.

Q. Do you have any final words for educators about multimedia?

This is the major challenge that multimedia now poses for educators—to bring all their artistry and creativity to the effort.

Sueann: There are two things. First, I'd like to invite academic faculty members to experiment with multimedia in their teaching and research. Second, I'd like to tell them to be sure to remember the element of "artfulness" in this field. To me, multimedia development has two sides—a science side and an artful side. At this point, I think that the technology is generally ahead of the art. This is the major challenge that multimedia now poses for educators—to bring all their artistry and creativity to the effort.

Current Multimedia Projects

The following are five case studies of multimedia projects under way today at various universities, illustrating everything from hypertext (nonlinear reading and writing implemented on a computer system for annotating and collecting ideas) to full-fledged multimedia environments (environments that extend the notion of hypertext to incorporate video, audio, and animation). They range from a hypertext environment used to teach proper research techniques (Project Jefferson) to use of multimedia in classroom demonstrations (at UCSF to teach techniques in hand surgery). Although some of the projects may seem at first glance a trifle esoteric, all have aspects that can apply to the use of multimedia technology in almost any field.

An Introduction to Project Jefferson

By Peter Lyman, Martha Boaz Library Research Professor
The University of Southern California

It is now possible—and urgent—to develop software that makes the information age an everyday resource for students.

As a newcomer to USC and the Project Jefferson software development team, two things struck me as remarkable: first, the diverse composition of the PJ planning group, and its conviction that it is now possible—and urgent—to develop software that makes the information age an everyday resource for students; and second, the use of HyperCard to integrate the use of diverse kinds of software while still respecting the individual learning style of the user. In this introduction to Project Jefferson, I'd like to focus upon the

problem that sparked this software development project, and the team dynamics that have driven it forward, as well as attempting a preliminary definition of Project Jefferson.

The group is a microcosm of the social movement that is academic computing—part academic, part technical, part commercial, and part visionary.

- Teachers—Rick Lacy and Irene Clark are writing teachers who experiment with the use of computers to teach freshmen not just how to write, but how to make research a way of life.
- Researchers—Mark Chignell is a professor of industrial and systems engineering, interested in designing systems for information retrieval, who brought evaluation into the core of the development process.
- Software developers—Tyde Richards is an information systems designer who develops new information technology for the university library; Dave Vronay is an undergraduate philosophy major with a genius for programming.
- Librarians—Margaret Johnson is a librarian using a mainframe as a network server with bibliographic databases and electronic research tools; Ann Lynch runs the USC College Library, where students come to use the software.
- Representatives of private industry—Susan Kinnell is from ABC Clio, and provides expertise on the nature and content of databases; Bernice Glenn is a consultant on interface design; and Lydia Zaffini is the campus representative of Apple Computer.

It is striking that the group has never been able to name the software it has written.

It is striking that the group has never been able to *name* the software it has written, still referring to it as "Project Jefferson," the name on the original grant application to Apple Computer. I think this reflects the nature of the group, the sense of grappling with an important problem that can now be addressed, but for which no particular software is *the* solution. Theoretically, the shared problem is that we live in a moment of transition, from an age in which knowledge resides in institutions (libraries, universities, classes) and is presented to individuals to an information age in which independent individualized learning is possible through electronic access to information. In this sense, the problem of Project Jefferson is to design software for an information age, software appropriate to the lifestyle of the autonomous learner. From an educational perspective, the Freshman Writing Program at USC wants to develop a tool to teach students to learn how to learn by using information technologies, not just how to do research to survive the last paper assignment.

The library is engaged in designing a new teaching library, and wants to learn how people use on-line services in order to design them better.

From a designer's point of view, Project Jefferson is an experiment with a hypertext environment in which different kinds of software can be summoned, and different kinds of information accessed, all within a consistent and individualized interface.

Project Jefferson is a self-contained research tool, thus a simulation that teaches students the skills to do research in the real world.

Finally, "the present version" of Project Jefferson is actual software (we always call it "the present version" because we're thinking of "the next version"). It is an adaptation of HyperCard to teach freshman writing students how to do research. It is a self-contained research tool, thus a simulation that teaches students the skills to do research in the real world (the *next* version will speak to networked databases). Its ruling conceptual metaphor is that it is *an electronic notebook* with which students can gain access to a paper assignment on the U.S. Constitution; read a dictionary or encyclopedia for background information on constitutional issues, and download key ideas to their electronic notebook; search a database of bibliographic information, taking notes if need be, or downloading a copy; and finally, dump all this information into a Microsoft Word file as the raw materials of a research paper. However, I think that the essence of Project Jefferson is that it is a computer disk that students carry around with them, a flexible tool that provides access to the software and information they need to learn what they need to know. In this sense, it is an effort to design software for a dynabook.

I have asked four members of PJ group—a teacher, a professor of industrial design, the head of the programming team, and a librarian—each to give his or her own view of the project.

The Perspective of the Software Development Team
Tyde Richards, University Library

We began development in May 1987. Our initial objective was to develop a citation database system so intuitively simple to operate that it could be used by freshmen to write their first research paper. Our grant commitment to Apple required that software be completed and on public display in USC's College Library by the EDUCOM meeting in October 1987.

Our initial development language was a compiled LISP, selected primarily for its "interface building" tool. By late July, we had a Boolean retrieval engine and a crude query builder to show for two months' effort. Work on important pieces of the software had not begun. At this time, we decided to switch to HyperCard.

The query builder, which had taken two months to develop, was converted into HyperCard in five hours.

That decision proved fortunate; we noticed an immediate and dramatic increase in our development pace. The query builder, which had taken two months to develop, was converted into HyperCard in five hours. Team members began experimenting with new features such as a "writing assignment" and a pop-up notepad. We soon realized that we were in a position to accomplish substantially more by EDUCOM than we had originally anticipated; however, we had no clear notion of what that "more" should be.

We took a big gamble on HyperCard by assuming that the two remaining months to EDUCOM gave us sufficient time not only to develop an application, but also to evaluate, redesign, and rewrite it. The evaluation prototype scrapped our initial design and added

new features, mostly related to writing. We identified the following key definitions of the software:

- a set of indexed citations
- a query builder based on a hierarchically ordered set of index terms
- definitions for the index terms
- a writing assignment
- assignment-specific prewriting questions
- a notepad

These features were implemented as six discrete contexts, with a shared menu that allowed the user to move between them. The prototype made no provisions for output.

Five weeks before EDUCOM, we froze the prototype and dedicated a full week to user evaluation. This involved about 30 evaluators—a diverse mixture of students, academics, and information-industry professionals.

We dedicated the following full week to system redesign. The evaluators' comments were paradoxical: they liked the prototype but had no idea what its purpose was. This raised a red flag: if the evaluators failed to comprehend the system, how could we expect any better from freshmen? The six major features in the prototype were all essential. What needed revision was not the prototype's features, but the way in which those features were organized.

The evaluators' comments were paradoxical: they liked the prototype but had no idea what its purpose was.

My solution to this problem was to split the application into two related yet distinct parts: an assignment-specific "notebook," residing on a student's personal disk; and a shared database or "locater," residing on a hard disk and applicable to many assignments. To solve the output problem, the contents of the notebook were to be converted into an ASCII text file, conceived of as a "protopaper."

We completed an initial version of the new design a week and a half before EDUCOM. It was minimal, ugly, and did not work reliably. Field-testing began immediately in pilot classes of freshman writers. Preparations were started to give the system a facelift for exhibition.

The exhibition revision began on the Friday before EDUCOM. We worked more or less continuously from that point on in a long and joyous madness. Each of the hundreds of cards in the system had its graphics changed. The icon set was finalized and integrated. The transition sounds were finalized and integrated. The system was pumped full of textual information. Index linkages were verified. Bugs reported in the field tests were analyzed and fixed. We completed the software at 5:30 in the morning on the first day of EDUCOM, installed it in the College Library, and went home to sleep.

The Teacher's Perspective
Rick Lacy and Irene Clark, Freshman Writing Program

Attempts to teach research paper preparation have been largely unsuccessful.

Recent recognition of the importance of writing and critical-thinking skills, both within and beyond the scope of the university, has focused attention on the role of the research paper in the composition course. However, attempts to teach research paper preparation have been largely unsuccessful. In response, the Freshman Writing Program at the University of Southern California has created a new model. It's called "the researched paper," and it enables students to develop critical thinking and research skills. Basically, before they are expected to grapple with and locate sources for a complex topic, they complete preliminary activities and assignments that are sequenced according to cognitive difficulty and the demands they place on student skills.

This model has been implemented through the creation of a topic-specific on-line information retrieval system that prepares students to incorporate sources into their papers in Freshman Writing and, ultimately, to use the full range of library resources. Project Jefferson presently consists of the following components: a word processor (Microsoft Word) to facilitate the generation, writing, and editing of texts; access to the on-line library catalog and specialized on-line databases (magazine and newspaper indexes); access to the Freshman Writing Program pedagogical database on the Constitution; and The Electronic Notebook.

The Electronic Notebook provides much the same function as a traditional notebook—a place for students to take notes or to "download" and store information.

The Electronic Notebook resides on the student's own disk. It provides much the same function as a traditional notebook—a place for students to take notes or to "download" and store information. The Electronic Notebook consists of the following sections: Focus Questions, Assignment, My Ideas, Background Material, and Citations. These sections correspond to the types of information that students create or transfer to their notebooks during their research.

Students generally work through the notebook sections in sequence: The Focus Questions section asks students questions to help them understand their own feelings and background on the subject of the assignment, the question of equal opportunity; student responses to these questions are written to a Microsoft Word file on the student's disk. Thus students can use freewriting and discussion in order to become preliminarily involved with the topic.

After working through the Focus Questions section, the student reads through the Assignment section, attempting to identify key terms about which he or she needs additional information, such as background material or definitions of unfamiliar words or concepts. Once students are aware of the key terms in the assignment, they can select these terms to access necessary additional information about the topic.

Moreover, the initial information screens they encounter are linked relationally to other information screens through the Related Terms field. This system of information screens and related terms defines a rich hypertext environment through which students can browse, following the relational pathways that they choose. As they browse, they can "photograph" any of the screens with the camera icon on the left side of the screen. This stores the information in the Background Information area of the notebook, where it may be viewed. At the end of the browsing, students need only click on the back arrow to return to the exact point at which they originally entered the hypertext information environment.

This system of information screens and related terms defines a rich hypertext environment through which students can browse.

Once students have gathered background information, they are then ready to search for secondary sources, which are indexed in hierarchical form. Each term in the index is then further subcategorized, modeling for students a method of organizing information. The Locator part of the software finds all articles in the database indexed for the term or terms chosen. Throughout the process, students can make notes and jot down ideas that occur to them, by selecting the section marked "My Ideas."

As a result of this initial search process, students accumulate a number of materials in the different parts of the notebook. Initial thoughts and general points of view will be found in the Focus Questions section. Subsequent writing (notes, jotted ideas, a rough outline, and so on) are in the My Ideas section. All "photographed" content information is in Background Information, and all citations in Citations. Cumulatively, these represent a substantial resource file. This file can be converted to a Microsoft Word file and, ultimately, incorporated into the paper as the student approaches the task of synthesizing information and writing.

Design Issues
Mark Chignell, Department of Industrial and Systems Engineering

Project Jefferson's current form arose out of the fortuitous combination of a group of people interested in on-line retrieval interfaces and a group of people who wanted to build a research component into the Freshman Writing curriculum at the University of Southern California. The major design modules involved the on-line retrieval interface, the on-line database to be used in the prototype, and the curricular materials that would motivate students to use the on-line database. The problems addressed by each of these modules were as follows:

- Information Retrieval:
  —Selecting articles and documents
  —Indexing the articles and documents
  —Building an on-line retrieval interface
- Curricular Materials
  —Constructing an assignment
  —Selecting (or writing) additional support materials.
- Evaluation:
  —Selecting instructors and classes to test ideas on.
  —Constructing methods for testing design ideas prior to proto type development.

It was initially assumed that database construction and indexing should be handled by a commercial database developer acting as a consultant. However, the database that was initially considered contained material that was too difficult for freshman usage. Similarly, the indexing had to reflect the needs and capabilities of the students. As a result, the database was hand built, and was generally subsumed under the curriculum development process. Meanwhile, indexing was carried out incrementally. First, a database expert constructed a hierarchy of relevant index terms. Then, a constitutional law expert refined this hierarchy. The result, however, was too technically oriented for freshman writing students and lacked detailed coverage of the more specific issues that would be addressed in a single assignment. Thus the final version of the indexing was carried out within the writing program.

One major specification for the software that remained unchanged throughout the development process was that the interface should embody a consistent model that assisted naive users.

One major specification for the software that remained unchanged throughout the development process was that the interface should embody a consistent model that assisted naive users (freshman writing students) in performing what would otherwise have been a difficult task. At first, a concrete model of constitutional controversies was proposed, based on the analogy of a case being tried in a court of law. However, questionnaires based on this model showed it to be poorly understood by students, even after a certain amount of training.

After further brainstorming, the concept of a student notebook was chosen as a metaphor for the interface. The notebook had the advantage of being a general metaphor that was familiar to students and that could be applied across a variety of instructional settings. The notebook metaphor also allowed the team to use dividers to separate out the different modules of the project in a natural way. Eventually, the notebook became divided into the five major sections already mentioned.

The Focus Questions and Assignment sections were based on the way that the freshman writing instructors typically introduced assignments to their classes. The Citation section reflected the on-line database interface that had been the initial motivation for the project. The remaining two sections arose out of the methods and orientations used in the project. As the notebook concept matured, it was recognized that there were two distinct types of notebook:

- The notebook that defined the assignment and the information that was generally available (the instructor's notebook).

- The student's notebook, which contained the assignment being worked on by the student. The student's notebook came to be seen as a customized version of the main notebook, which included those documents in the database that the student thought to be relevant, along with notes, annotations, responses to the focus questions, and so on. Once the distinction between the two types of notebook became clear, the team was faced with the issue of how to transfer information from the instructor's notebook to the student's. The metaphor of photography was chosen, because it represents a familiar way of copying information. By taking photos, students could copy information and make it available in their own notebooks.

The metaphor of photography was chosen, because it represents a familiar way of copying information.

Once the concept of the student notebook was developed, it was a short step to collecting the screen photographs, citations, notes, and annotations into a text file (the "protopaper") that would form the basis of the student's final report. The construction of the notebook marked a transition, because it served to integrate the disparate ideas and tools that we had been working with previously.

We needed a method that would allow students with little or no research experience to select index terms as part of the query. We solved this problem by building a menu of index terms into the online retrieval interface. As early as fall 1986, we had decided on a hierarchical organization of the index terms. Once we built this hierarchy, we still had the problem of making sure that students understood these index terms well enough to use them in building search queries. Our solution was to link a hypertext representation of the index terms—complete with annotations, definitions, and links to related topics—to the index hierarchy. This hypertext could then be used in two ways: by browsing through background information or by finding definitions and information relating to specific search terms.

October 1987 was a major landmark for the project: its first public demonstration, at the EDUCOM conference in Los Angeles. It also marked the beginning of the redesign stage. This redesign is occurring against the backdrop of the software actually being used in freshman composition classes at USC. In fall 1987, approximately 70 freshmen used the software. By spring 1988, this number had climbed to 200, and the project already represents one of the largest computer-based instruction field studies of its kind yet attempted.

The project represents one of the largest computer-based instruction field studies of its kind yet attempted.

Our initial goals have largely been met, but there are a number of deficiencies in the current software. First, there are only two assignments currently available, and the creation of new versions of the notebook is quite a chore. Second, we do not yet have good models and tools for determining what effect this software and instructional environment is really having on the way that students

learn. Third, we do not have the sophisticated documentation or on-line help facilities that might be expected of an expensive commercial implementation of this functionality. And finally, we do not yet allow notebooks to be linked to mainframe databases that may contain millions of documents. Each of these issues represents a fertile area for research and development for the Project Jefferson team and for the disciplines of computer-based instruction, freshman writing, and on-line retrieval.

The Perspective of the University Library
Margaret Johnson and Peter Lyman, University Library

Staff members of the University Library have participated in Project Jefferson in order to learn how to design the library of the future.

From the beginning, staff members of the University Library have participated in Project Jefferson in order to learn how to design the library of the future. The library staff needs to learn how to design the interfaces with which users can search large on-line bibliographic databases, and needs to learn how to teach new kinds of research skills to library patrons. Project Jefferson is a significant experiment on both grounds.

Although the initial focus of Project Jefferson has been to teach research skills in freshman writing, in principle, it is an example of an integrated tool that a researcher in any discipline could use to search for information resources, develop a bibliography, annotate it, and copy it to a file on a personal disk. This is the goal of the next version of Project Jefferson software—a generic library research tool designed to access networked information resources of many kinds.

Project Jefferson is also a joint experiment with the Freshman Writing Program to teach students the skills necessary for library research. In the present version, students construct their resource files within a highly supportive, topic-specific database that has been constructed and indexed by the faculty of the Freshman Writing program. This hypertext version of bibliographic research is really a simulation of library research; we do not yet know if the skills learned in a simulated environment form an adequate conceptual model for research in a real library.

The University Library is now using one of USC's IBM mainframes as a network server that provides "USCInfo," an on-line catalog and an on-line index to databases of citations in periodical articles (Magazine Index, Trade and Industry Index, Computer Data Base, Management Contents). As the students' skills improve, it is expected that they will move into this "real" world of on-line databases and catalogs, carrying with them the skills that have been modeled and developed earlier.

By testing this software in USC's College Library, a setting without the consulting support provided in the Freshman Writing Laboratory, we can test the extent to which the software really is "intuitive." We can also learn more about the kinds of teaching skills that librarians will have to have in the state-of-the-art undergraduate Teaching Library that will soon be built at USC.

However, the transition will also be eased by changing the "real" databases themselves to become richer, more supportive environments. Project Jefferson is an opportunity to learn more about how students do research; the skills needed for research; the skills needed for end users to do unassisted searching; the training programs that need to be developed to assist in providing access to on-line information. From the library's perspective, Project Jefferson is a laboratory for experimenting with more intuitive and powerful user interfaces. Are there other ways to organize information that would provide better access? How can new interface designs be integrated into the ways in which abstracting and indexing services assign subject terms or descriptors?

From the library's perspective, Project Jefferson is a laboratory for experimenting with more intuitive and powerful user interfaces.

A note on the future:
Project Jefferson software will be available through Kinko's Academic Courseware Exchange in the fall. USC would like to form a consortium of schools—to test the software in the classroom to help guide future development, and to create and exchange bibliographic stacks on a variety of subjects appropriate for the teaching of writing and research skills.

For more information, please contact:
Tyde Richards
Doheny Memorial Library
The University of Southern California
Los Angeles, CA 90089-0182

Brown University Intermedia Project

Intermedia, developed by Brown University's Institute for Research in Information and Scholarship (IRIS), is a general-purpose environment in which "idea workers" can compose complex, sophisticated webs of multimedia information. It provides multiple applications and mechanisms to link the contents of documents created with those applications. It is both a reader's tool and an author's tool, integrating the creation of materials and the following of connections into a single seamless multiuser environment. The system is now entering its second year of testing in several Brown University courses, including Plant Cell Biology taught by Professor Peter Heywood and A Survey of English Literature taught by Professor George Landow.

Intermedia is a general-purpose environment in which "idea workers" can compose complex, sophisticated webs of multimedia information.

Intermedia is considered a *hypermedia* system. In essence, a hypermedia system allows authors or groups of authors to link information together to create paths through a body of related material, annotate existing texts and illustrations, and create notes that direct readers to bibliographic data or the body of the referenced text. Using a computer-based hypermedia system, students and researchers can quickly follow trails of footnotes and references without losing their original context; thus, they are not

obliged to search through library stacks to look up referenced books and articles. Explicit connections—links—allow readers to travel from one document to another, effectively automating the process of following references in an encyclopedia. In addition, hypermedia systems such as Intermedia that support multiple users allow researchers, professors, and students to communicate and collaborate with one another within the context of a body of scholarly material.

The system, which runs on a network of UNIX-based Apple® Macintosh® II computers or IBM PC/RTs with specially licensed Macintosh Toolbox routines, currently contains five integrated applications: InterText, a text professor; InterDraw, a graphics editor; InterPix, a scanned image viewer; InterSpect, a three-dimensional object viewer; and InterVal, a timeline editor that allows users to interactively organize information in date and time sequences. These applications conform to Macintosh/Microsoft Windows interface standards. Any number of documents of different types, along with the folders containing these documents, may be open on the desktop at one time. In the future, a two-dimensional animation editor, a map editor, links to dictionary entries, searching of remote databases, and videodisc access that allows users to interactively create sequences of clips from a videodisc for subsequent viewing will be added to the system.

Navigational linking is the major paradigm of Intermedia.

Navigational linking is the major paradigm of Intermedia. To link information, one picks a source selection in a document, chooses the start Link command, indicates a destination selection, and gives the End Link command. When the operation is finished there is a semantic tie, a navigational link, between the source and the destination. Whenever the user selects a link symbol and issues the Follow command, the document containing the other end of the link is activated and the appropriate information is presented in another window on the screen; the document is retrieved from storage if necessary. In the near future, users will be able to ask for categories of links so that only link symbols that meet the user's criteria are shown on the screen. Such organization will allow an individual user, a group of users, or in the extreme, an entire campus to create a shared web of materials.

Using Intermedia on a network of computer workstations, students can electronically complete assignments that include links to a variety of references. For example, a student preparing a report on the Venetian Cinquecento in an art history course might include links to a chronological view of the political and cultural events of the period, biographies of noted artists, diagrams illustrating compositional theories, and video reproductions of relevant works in his/her document. The instructor (and other students, if permitted) could read the document, examine the reference material, and add personal annotation links such as comments, criticism, and suggestions for revision. While revising the document, the student could see all of the instructor's comments and examine the sources containing counterarguments.

Intermedia has been developed with funding provided by IBM Corporation, Apple Computer, and the Annenberg/CPB Project. For additional information and/or participation in IRIS's industrial affiliates program, please contact IRIS Associate Director Norman Meyrowitz at (402) 863-2943.

For more information, contact:
Nicole Yankelovich
Project Coordinator
Institute for Research in Information and Scholarship (IRIS)
Brown University
Box 1946, 155 George Street
Providence, RI 02912

New York University Computer-Interactive-Video Project

Background and Aims

The Computer-Interactive-Video project is a schoolwide multidisciplinary effort to provide medical students with a unique interactive environment for small-group exercises, independent study, and self-assessment. The project began in the spring of 1987 with four faculty members and five medical students. As of this writing, there are 22 faculty members, representing 18 disciplines, working on various learning modules.

The Computer-Interactive-Video project is a schoolwide multidisciplinary effort to provide medical students with a unique interactive environment for small-group exercises, independent study, and self-assessment.

Each faculty member works with one student; together, they constitute a development team. The faculty member's role is to decide on the aim of the module, its overall presentation format, and its content. The student acts as the liaison to the computer (our faculty, for the most part, is not computer literate). In addition, students provide continual feedback to the faculty, and bring fresh perspective and new ideas to the project. This arrangement has been extremely beneficial, both in designing material and in expediting development. Fallout from this joint effort has been the creation of a cooperative atmosphere highly conducive to an open dialog on rethinking and restructuring medical education at our institution.

Since the medium is relatively new, and since we do not know what works and what doesn't, a variety of pedagogical designs have been employed. Modules run the gamut from self-test review sessions to exploratory hypermedia presentations.

This spring, learning modules have been used for the entire first-year class in histology and microbiology. The initial response has been overwhelmingly positive. More quantitative evaluations are planned as more material becomes available.

Vital Statistics

We are presently using Macintosh SE and Macintosh II computers. We chose the Macintosh because we had noted that our students and faculty seemed to migrate to these machines when they had a choice. In addition, the graphics capabilities of the machines fit our need for presenting information graphically rather than as text.

Video images are captured using WORM (write once, read many) technology on a Panasonic videodisc recorder (TQ-2026F) and Sony (DXC-3000) and JVC (BY-110) video cameras. Microscopic images are captured through a Nikon microscope and JVC (BY-110) camera setup. Projection slide images are obtained using a Tamron Fotofix video film recorder. Video displays employ Panasonic players (TQ-2027F) and Sony monitors. Digitizing instruments include MacVision and Dest scanners.

The authoring languages we are currently using include HyperCard (Apple Computer), Video Builder (TeleRobotics), and VideoWorks II (MacroMind). Interfaces for the Panasonic players to the Macintosh systems are via drivers for HyperCard or Video Builder software.

Plans

We are finishing the first phase of the project—the early developmental stage—and have entered a time when students need to have a facility to work with the modules. We expect to spread clusters of workstations around the medical center in teaching locations, the library, and student dormitories.

Resources

The recent explosive growth in the number of participants has indicated a need for at least one full-time staffer.

Funding for the project has been internally generated. Funds for the next phase of the project are being sought from private foundations. So far, we have used only students and faculty. However, the recent explosive growth in the number of participants has indicated a need for at least one full-time staffer to man the development facility, maintain the equipment, and help train participants. The disciplines represented include Anatomy, Histology, Immunology, Microbiology, Molecular Biology, Neurosciences, Parasitology, Pharmacology, Physiology, Pathology, Rheumatic Disease, Endocrinology, Oncology, Nephrology, Cardiology, Respiratory Disease, Obstetrics and Gynecology, and Radiology.

Dissemination of Material

It is our intention to share whatever material we can with interested institutions and individuals. Some stand-alone modules will be offered through Kinko's Academic Courseware Exchange. Others may be obtained through direct requests.

For more information, contact:
Martin Nachbar, M.D.
Project Coordinator
New York University Medical Center
550 First Avenue
New York, NY 10016
(212) 340-5744

Hypertext and "Intertextuality" at the College of Wooster

Hypertext

First suggested by Doug Englebart in the 1950s, the idea of hypertext has been around nearly as long as Turing's idea of a universal machine. Englebart envisioned a computerized system interconnecting a library of materials—textual and graphic—electronically. With hypertext, instead of trudging from study to card catalog to library stack in search of a referenced text, one simply pushes a button or touches a word on a screen. Instantly, a relevant text or image—or both or several—pops into view, or, perhaps, a menu of related materials appears.

With hypertext, instead of trudging from study to card catalog to library stack in search of a referenced text, one simply pushes a button or touches a word on a screen.

However hypertext is actually realized (and there are several working examples now), it is a set of powerful tools that allow for faster connection among texts than has ever been possible. Where before one was directed by note to one or more related texts selected by an author, hypertext presents the actual texts and pictures themselves. Moreover, the texts and pictures that pop up in reference to a beginning text are themselves rich with further references. The clear implication is that there could be one, final, constantly growing hypertext, connecting every lexical and graphic text ever produced. One would begin, perhaps, by connecting all the texts in the Library of Congress and all the images in the National Gallery.

Hypertext simulates "intertextuality," the process through which a reader makes meaning by combining the codes of the reader's culture, learned codes relevant to the culture of the text at hand, and the codes the reader infers from the text at hand. Hypertext allows all of the texts that represent these codes at work to be physically connected to one another.

Hypertext in the Curriculum

The College of Wooster is exploring the use of hypertext implemented on the microcomputer in its classes. In a course that will be taught in the fall, students and faculty will use Owl International's Guide to create hypertexts as a way of exploring the linguistic and cultural codes present in late 17th-century England.

The course centers around a single year, 1667, and a single day, March 2.

The course centers around a single year, 1667, and a single day, March 2. On that day, surveyor-general of the victualing office Samuel Pepys spent the morning planning purchases for the English Navy. After dinner, he and his wife, Elizabeth, traveled to the Theatre Royal, where they saw John Dryden's *Secret Love,* or *The Maiden Queen.* Pepys was captivated by the performance of Nell Gwyn as Florimel and, eventually, saw the play three more times. On this same day, John Milton, living on Jewin Street just beyond the city walls of London, was searching for a publisher for the epic poem *Paradise Lost.* Aphra Behn, a 27-year-old woman in the employ of the court of Charles II, was at this time in Antwerp as a spy for the government. One of the more noticed books of the year, Thomas Sprat's *History of the Royal Society,* was of particular interest both to Pepys, a member of the Society, and to Charles II, who had founded it in 1660. Sprat's book defended the new science being promulgated by Bacon and proposed a new diction and style of writing appropriate to it.

The course will begin with five texts:

- The March 2 entry in Pepys's diary
- Dryden's *Secret Love*
- Aphra Behn's letter to a friend, denying that her gender hindered her effectiveness in business and politics
- Selections from *Paradise Lost*
- Selections from Sprat's *History*

The course will consider how the five texts, inevitably related to past texts, become, in turn, embedded in subsequent texts.

Starting with these texts, students will be asked to begin acquiring the codes—the language, sign-systems, and cultural assumptions—that enabled the formation of these texts and whose relationships to them yielded various meanings for individual readers. The attempt to reconstruct these patterns will call for the reading of earlier texts in order to determine some of the symbol systems available. Eventually, the course will consider how the five texts, inevitably related to past texts, become, in turn, embedded in subsequent texts.

As an example of a set of relationships, one may note that in Dryden's *Secret Love,* the characters portrayed by Nell Gwyn and Charles Hart establish a code or convention utilized and embedded in comedies well into the next century. The power of this code may be suggested by some of Aphra Behn's subsequent plays. Because she earned an independent living as a government employee and, later, as a freelance writer, Behn's life gave the lie to conventional notions about women's limitations. Yet in her plays, the literary codes of the coquette, and certain social codes concerning the roles of women, overpower her own experience.

The instructors will suggest various links between texts and culture, always explicitly indicating that these are only a few of the ways of creating order and encouraging students to discover others. A central, stated hypothesis will be that meaning arises from the interaction of symbol systems from the full range of experience—from science, religion, politics, and daily life, as well as from literature.

This example is presented here as examples will be presented by the instructors in the course. However, the heart of the course will be the relationships among texts and images constructed by students. Much of the business of the course will involve students and instructors in analysis and criticism of the principles by which they organize such constructions.

The heart of the course will be the relationships among texts and images constructed by students.

These relationship sets will be simulated using Guide, which allows the embedding of a text—of any length and in a variety of formats—within another text. In the convenient jargon of the program, combinations of text with built-in links are called Guidelines.

Course Design

Weeks 1–2: Instruction in the use of Guide, discussion of theories of intertextuality, and initial discussion of the five texts with which the course begins.

Weeks 3–4: Each student uses Guide to make connections among the five texts and any other texts. Class time is spent sharing the kinds of connections that students make. Discussion will often probe for the assumptions that underlie the order students have given their Guidelines.

Weeks 5–7: Several texts will be presented for which the instructors have prepared Guidelines. Discussion will concern the assumptions on which these were prepared, the kind of order they impose, and the possibilities for very different assumptions. Students will be contributing other links and arguing for their assumptions.

Week 8: Students will selecting the texts that they will use for their major project, the creation of a hypertext. They will also be articulating the set of assumptions on which they will select particular kinds of links as against others.

Weeks 9–12: Students will be creating their hypertext. Class time will be divided between the sharing of hypertexts and the discussion of 17th-century texts.

Weeks 13–14: Class time will be devoted primarily to the combining of the individual hypertexts into one huge hypertext (perhaps on the VAX if we have the Mac-VAX connection equipment by then). Out of class, students will be working on a final "hyperpaper." This will be an electronic document, created using Guide, in which students discuss the process by which they have created their hypertext and the assumptions on which it was based. In doing this, students will necessarily discuss their understanding of intertextuality and of cultural codes in the 17th century.

Along the way, instructors will assign the reading of a number of 17th-century and earlier texts. These will include poems, plays, fiction, diaries, letters, and philosophical, religious, political, and

scientific essays. The instructors will make clear their own assumptions concerning various links among these texts and will encourage students to make their own links. Students will be required to go beyond the assigned reading in the creation of hypertexts.

The Theory

The project invites students to imagine and test the rules that govern intertextuality—"the multiple ways in which any one literary text echoes, or is inseparably linked to, other texts, whether by open or covert citations and allusions, or by the assimilation of the formal and substantive features of an earlier text, or simply by participation in a common stock of literary and linguistic procedures and conventions" (M.H. Abrams, *A Glossary of Literary Terms*)—as they construct hypertexts. Our course will bring a delicately nuanced theory heretofore to be abstracted only from dense critical anthologies and quaintly titled journal articles (many of them in French) to the level of trial and error.

Hypertext is a model of the way texts interrelate. But like all good simulations, it is also a superb stimulus.

It will be trial and error only within the simulated world of hypertext, of course. Hypertext is a model of the way texts interrelate. But like all good simulations, it is also a superb stimulus. As they play with a computer program, students will be grappling with at least the following concerns:

- Which texts fit with which other texts?
- Where do they fit—where should the actual connection between text A and text B be placed—at which word or phrase or space between words?
- In theory, there is only one, final hypertext, but in the new computer classroom in Taylor Hall, with Guide software and Macintosh personal computers, we can make only pieces. Are there any limits on the kinds or numbers of texts that fit immediately together? Are those limits rule-bound?
- Is the ultimate hypertext one in which all texts and pictures in the universe are referenced to every written word, so that one could actually see the world in a grain of sand? Or would the ultimate hypertext be organized chronologically? By Library of Congress or Dewey Decimal topic? By Jungian archetype? By myth? By socioeconomic class or sexual orientation? By relation to instinctive need?
- In what ways are the hypertexts made by students and faculty in the course good representations of the way in which texts are actually formed?

Because structuralism guides theory and research in a variety of fields, hypertext has a huge range of applications. The obscurity of most of the 17th-century texts we will deal with in this course is a sign of the method's usefulness in simulating any meaning-system. One could construct a hypertext of Kachina dolls or voter choices or kinship group members or fashion.

In every case, the hypertext would actualize a way of connecting meanings. There would be active learning both in the construction of the model and in its criticism.

The comments above are excerpts from our proposal for a new English course utilizing hypertext as a way of simulating intertextuality. The proposal has been accepted by the Department of English and by the Vice President for Academic Affairs, and the course is to be taught in Wooster's computer classroom this fall.

For more information, contact:
Peter L. Havholm (216) 263-2243 or
Larry Stewart (216) 263-2412
The College of Wooster
Wooster, OH 44691

Macintosh and Medical Instruction at UCSF: A Profile of Bob Markison

The office is small and cluttered, with an overwhelming array of computer equipment. The back wall, dominated by bright green blinds, houses a Microtek scanner, a SuperMac monitor, a monochrome monitor, a Macintosh II, a VCR, and a ProViz digitizer. To the left are several overflowing bookshelves, with computer manuals and art books mixed with cans of adhesive, glue, and videocassettes on airbrushing. A cabinet stands against the right wall with drawers marked for overlays, transparencies, watercolors, and drawings. A beautiful, sensuously rounded sculpture sits in one corner. Cool jazz fills the room from a Walkman hooked up to two Bose speakers.

A man enters the room—medium height, stocky, balding black hair shot through with gray—sits in front of the computer, and begins to input data. Suddenly, his beeper goes off. You hear a voice over the static saying extension 8851. He quickly punches in the numbers and says, "Bob Markison."

An artist with a beeper? Well, yes and no.

An artist with a beeper? Well, yes and no. Bob Markison (featured in our cover photo) *is* an artist and musician, in addition to being cofounder of the UCSF Health Clinic for Performing Artists, chief of hand surgery at SF General Hospital Trauma

Center, and an assistant professor of surgery at UCSF, specializing in the redesign of musical instruments for the treatment of injured musicians.

It is Markison's ability to incorporate so many different fields into one schedule that makes him unique. But he sees no alternative: "My first language in life was music. The second was verbal. The third was surgery. The fourth is sculpture, and the fifth is art and animation." Markison could easily be a computer jock working away in some ivory tower, or just as easily, a professional musician or artist. Yet it is the combination of these five elements that makes him complete.

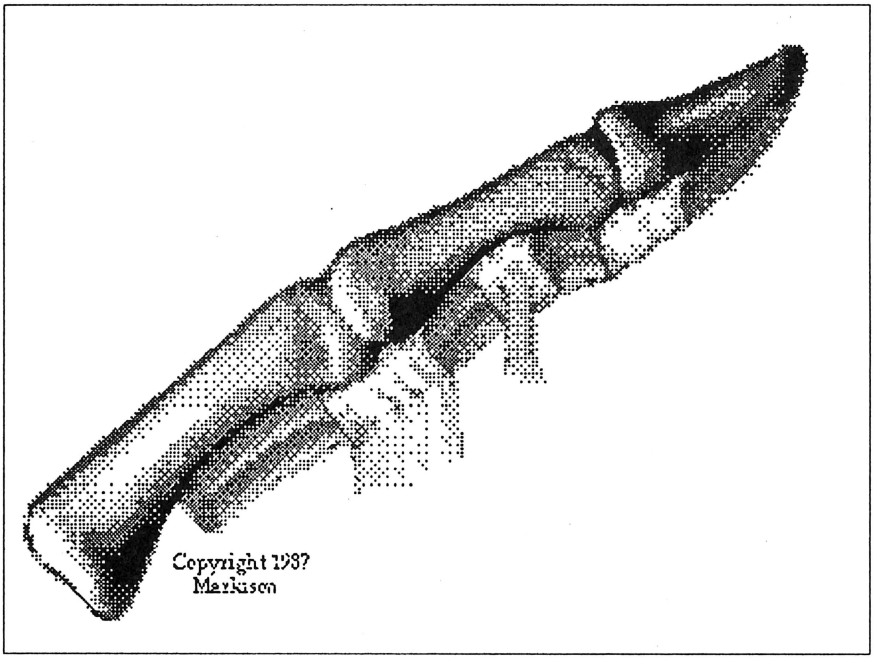
The following article by Markison gives an excellent overview of how he has incorporated the Macintosh into his teaching.

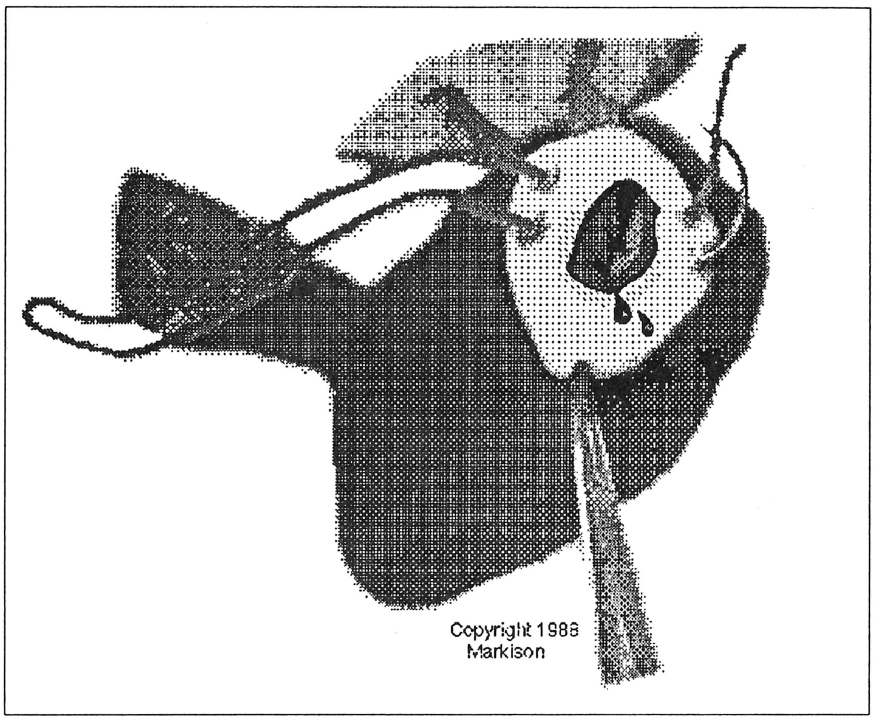
A Mac in the Hand is Worth....
By Bob Markison

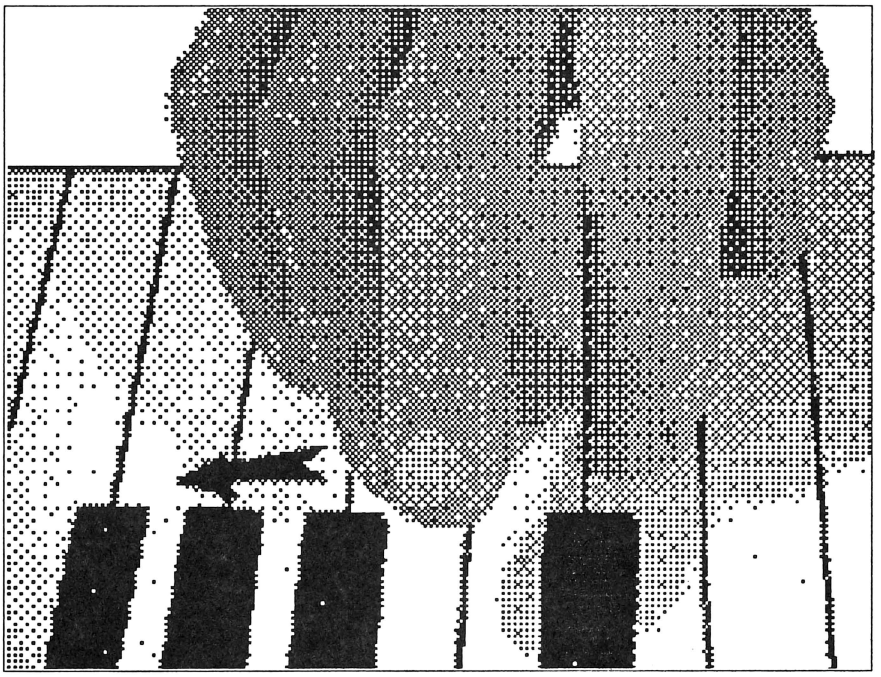
Introduction

The interface of the Macintosh is just right for teaching a visual subject like the surgical and nonsurgical care of hands.

The interface of the Macintosh is just right for teaching a visual subject like the surgical and nonsurgical care of hands, and it has occupied a prominent position in my teaching program for more than three years. This article describes use of the Macintosh for teaching basic hand anatomy and care of injuries; sorting out the often subtle problems of musicians' hands; rapid production of medical illustrations for publication; and creation of presentation materials—both still and animated—for audiences of various sizes.







The Hand

The hand is the most broadly represented organ on the sensory and motor areas of the brain. A view of a geographic map of the surface of the human cortex reveals that the territory assigned to reception of sensory signals and integration of muscular activity in

the hand is far greater in size than the regions responsible for the control of "automatic" organs such as the heart, lungs, liver, and kidneys. Hand injury states are therefore inextricably linked to conscious and subconscious mental process. Sometimes apparently minimal injuries will evoke profound changes in a patient's mood and outlook.

The hand is the most commonly injured body part. Of emergency-room visits in the United States, 20 percent to 30 percent relate to problems of the hand or upper extremity. Hands work around dangerous tools, pet unpredictable animals, express rage in fistfights, pare fruits with sharp knives, cushion our falls, and so on. They are also the means of livelihood in some capacity for most people.

Hand injuries are also the most costly injury category. Each year in the United States, $20 to $30 billion is spent for the care of injured hands. This includes the cost of medical care, time away from work, vocational rehabilitation, and legal expenses for injury-related lawsuits. Hand injury costs twice as much as head injury.

Ironically, hand anatomy and the diagnosis and treatment of acute and gradually acquired hand problems are taught minimally in most medical schools. The average medical student in most countries spends a day or two, seldom a week, on this very complex organ. Although the speciality of hand surgery has matured nicely over the years, the average nonspecialized physician involved in acute care may have a modest knowledge base for the care of this most commonly injured body part. Memorable Macintosh-generated teaching materials are proving highly successful in preparing developing physicians to manage hands properly.

The Hardware
Macintosh II with 5 megabytes of RAM
Jasmine 80 hard disk
Jasmine megadrive using 10-megabyte floppy disks
Mouse
SuperMac 19-inch monitor
ProViz video digitizer
Microtek MSF-300 flatbed scanner
Lighted rotating stage (custom-built)
High-resolution black-and-white camera
Color video camera
VCR
Julian Systems' NTSC converter for videp projection
35mm camera
Artograph opaque projector
Custom-built rear slide projection unit
MagicArm (articulated arm to hold 35mm camera)
Plaster casts of hands in various positions
Pencils, crow quill pens, and other conventional art supplies

Graphic Resources
Hands: live hands, plaster casts, cadaver dissections videotapes, 35mm slides and black-and-white photographs
Hand-drawn pencil, pen-and-ink, watercolor and airbrushed illustrations
Plasticine clay models of surgical maneuvers

Music Production
Musical instruments: real and digitized band and orchestra instruments
Sheet music and Macintosh-composed music
Yamaha WX7 MIDI Wind Controller Instrument
Yamaha TX81Z synthesizer
Opcode Studio Plus 2 MIDI interface

Music Software
Performer 2.3
Professional Composer

Uses

Teaching medical students, residents, and colleagues about hand surgery is easy with the Macintosh. When using a Macintosh Plus with earlier Finder versions, the following simple resources can be used. The workhorse for this type of teaching is Slide Show Magician, in which MacPaint text and digitized (or drawn) graphics are used for teaching concepts related to management of a given patient's injury. Each segment of a "show" begins with a wound "painted" onto a digitized hand, accompanied by a series of management questions (see figure). The discussion is enlivened by the use of the mouse cursor by students to draw in the vulnerable structure on the screen. Subsequent slides show the deeper structures of the hand by way of digitized original drawings and single frames of operating-room videotapes captured by ProViz. I am present to draw on the computer and on the blackboard in order to amplify the concepts involved.

The introduction of HyperCard allows for a more interactive approach to teaching. It permits the use of bit-mapped images in a branching, student-directed hypertext approach to teaching and learning. These HyperCard stacks include linkage to full-screen animation in color by use of the VideoWorks HyperCard driver. Ultimately, the memory requirements of these materials will require the use of a CD-ROM drive or laserdisc. At present, this material is being mastered for that purpose.

The introduction of HyperCard allows for a more interactive approach to teaching.

I also frequently use VideoWorks animations for the demonstration of surgical techniques, such as placement of sutures for tendon repair. These are either drawn freehand or prepared from digitized clay models and/or frames of operating-room videotape. (My background in animation comes from courses in cell and clay animation taken in the Film Division of S.F. State University. My art background is from courses taken at the UC Berkeley Extension campus in San Francisco, and other art classes given in the Bay Area. I feel strongly that all computer endeavors should be supported by training in the conventional [noncomputerized] modes of producing the same material. The use of PageMaker 2.0, for example, could be preceded by reading Adrian Wilson's *Design of Books* and *Pasteups & Mechanicals* by Demoney and Meyer.)

The VideoWorks animations have been enhanced by the use of music. I use the following procedure:

1. Composition of the musical score with Professional Composer.
2. Individual playing of the score parts on a WX7 set to the appropriate musical voices through the TX81Z synthesizer via the Opcode Studio Plus 2 MIDI interface into the Macintosh.
3. Sequencing and quantizing for the note sequences in Performer.
4. "Playing" of the musical sequence out through the TX81Z with a mixer attached onto cassette tape.
5. Selective sampling of the cassette master via MacRecorder into the Macintosh.

6. Sound editing with the Farallon SoundEdit software.
7. Pasting edited soundtracks into VideoWorks using VideoWorks sound-to-video software.

The Macintosh shines as an "electronic sketchpad" for brainstorming ideas.

Sorting out problems of the musical hand is assisted by the Macintosh in several ways. Since problems of musicians often involve difficulties relating to the hand and the instrument as a system, it is often helpful to consider how the instrument can be modified, as well as the hand and its technique. Sometimes this involves the complete redesign of the instrument itself. When this is necessary, the Macintosh shines as an "electronic sketchpad" for brainstorming ideas within a set of digitized views of all the band and orchestra instruments. The actual redesign of musical instruments for troubled hands then flows from scaled drawings of computer-modeled key reconfigurations, which the patient can take to the instrument maker so that prototypes of reversibly modified instruments can be made.

Medical illustration is also vastly simplified by a combined use of the Macintosh and conventional artist's tools. Production of a finished illustration for desktop publications using the Linotronic is accomplished by the following sequence of steps. A preliminary source, such as a video frozen frame through ProViz or an image from the Microtek scanner, is brought into the Macintosh. This serves as either a line drawing rendered in Adobe Illustrator or a preliminary gray-scale image that is put into finished form in ImageStudio. ImageStudio-edited gray-scale renderings can be colorized selectively in Pixel Paint for inclusion in VideoWorks animation, or they can be used to make slides by shooting the screen image with a 35mm camera. (Soon, an improved and affordable high-resolution film recorder will be available for the Macintosh from Mirus Corporation). A screen shot is made of the final image, or it is added to text in a page layout for camera-ready production and sent to the publisher on disk.

Presentation materials are greatly enhanced by use of the Macintosh. HyperCard is used for small-group teaching (as noted above), and printouts of the Paint documents are distributed to the class for note taking. VideoWorks is used either on the Macintosh itself or, for larger audiences, a videotape is shot of the SuperMac screen, and a soundtrack, generated by the techniques described earlier, is then dubbed onto it for video projection in an auditorium. Recently, we have had fine results with projecting the Macintosh signal through the Julian Systems NTSC converter box to a video projector. Black-and-white slides are made by use of a copystand-mounted 35mm camera loaded with Kodak LPD 4 direct-positive high-contrast copy film, which is then developed in an X-o-mat (X-ray developing machine).

Future Directions

Future directions include practical implementation of artificial intelligence and expert systems in three-dimensional animation created on more advanced systems, which can then be ported to the Macintosh. Happily, third-party developers are realizing the value of the Macintosh II, with its newfound connectivity, in sophisticated media of this type.

Future directions include practical implementation of artificial intelligence and expert systems in three-dimensional animation.

(I feel that it is essential (in our age of copyright infringement) to create all material—musical, graphic, and text-related—afresh. A premium must be placed on original ideas in all multimedia productions.)

Conclusion

A Macintosh in the hand is worth far more than just a set of digitized digits!

Reprinted (slightly edited) by permission of BMUG from the spring/summer issue of its newsletter.

Multimedia Feedback: Share Your Hopes and Concerns

What do you think? How will multimedia affect teaching? Research? Learning? Curriculum? If you will share your ideas, we will put together a summary of all comments received and make it available to all who are interested.

What do you think? How will multimedia affect teaching?

Kristina Hooper has listed several themes that may stimulate your thinking, and I have added some others. Please feel free to send us your ideas on any or all of these (or add your own). Send all material (preferably on disk) to Peter Olivieri, Wheels for the Mind, Boston College F430, Chestnut Hill, MA 02167.

Multimedia in Education

- What is the nature of interactivity? Are linear presentations obsolete?
- What can be done with all the imagery made possible with videodiscs and the sounds enabled by compact discs? Is any of this really new?
- Are emotional presentations appropriate to education? Or should technology be used primarily to present "factual data" in educational settings?

- Is the central design of educational presentations to link large amounts of information? Or is it to "tell stories"? Should presentations emulate libraries (act as reference centers) or classrooms? Is there a significant difference between encyclopedias and textbooks and TV documentaries, novels, and lectures?

- What balance is required between directed search and browsing paradigms? Can one count on individuals to bring their own questions and research strategies to their tasks, or must the technological environments provide these?

- What will it require to make the presentations now quite familiar in research laboratories available to the general community? Why is it seeming to take so long?

- What do we know from cognitive theory to mold our efforts in designing with new multimedia interactive systems?

- How will multimedia educational "stations" fit into schools—or will they?

- How can we best develop a language to describe the "experience" of a technology-based presentation? How can we both focus on the hardware/software required for the delivery of this experience and be sure that the experience is the primary motivator and criterion for developments?

- Will there be a standard interface to multimedia experiences—and should there be? If so, what is the appropriate metaphor for this interface? Is it the familiar desktop? The understandable spatial map? A task-oriented environment? A set of research tools? A number of construction kits?

- What happens when you combine the traditions of moviemaking, graphic design, computer education, encyclopedia development, text publication, public television, computer workstation design, classroom teaching, library organization, entertainment, and psychology? Will the results be significant educationally?

- Can learning be entertaining—and should it be?

- What will the instructor's classroom workstation look like in 10 years?

- What will the typical classroom look like in 10 years?

- What will the Macintosh computer look like in 10 years?

If a mere 100 of you respond, we will have some great things to share.

As you can see, there are some thought-provoking questions. Jot down your reactions. Write us a paragraph. Write 10 pages. Prepare a thesis. Most of all, share with others who are equally committed to teaching and learning. If a mere 100 of you respond, we will have some great things to share. Don't leave it to someone else—send your comments in now. Thanks in advance!

Multimedia Resources

There are a variety of resouces that may be of interest to those who are becoming involved in the implementation of multimedia in learning. The new *Apple Guide to Authoring Tools* contains a large reference section. If you have additional suggestions, please send them along so that we can share them with others.

CD-ROM: The New Papyrus
Steve Lambert and Suzanne Ropiequet, Editors
Microsoft Press
$21.95
619 Pages

CD-ROM: Optical Publishing
Suzanne Ropiequet, Editor
Microsoft Press
$22.95
358 Pages

CD-ROM Review
The Magazine of Compact Disc Data Storage
Subscription Services
P.O. Box 921
Farmingdale, NY 11737-9621
$19.97/yr (12 issues)

Interactive Multimedia
Sueann Ambron and Kristina Hooper
Microsoft Press, 1988

The Media Lab
Stewart Brand
Viking Press
40 West 23rd St.
New York, NY 10010
$20.00
285 Pages

The Society of Mind
Marvin Minsky
Simon & Schuster
$19.95
338 Pages

AV Video
25550 Hawthorne Blvd.
Suite 314
Torrance, CA 90505
$25.00 year (12 issues)
(213) 373-9993

T.H.E. Journal
Technical Horizons in Education
Information Synergy, Inc.
2626 S. Pullman
Santa Ana, CA 92705
Free to qualified subscribers.

Interactive Videodiscs
A Directory for Educators
Merrimack Education Center
Chelmsford, MA 01894

A Practical Guide to Interactive Video
Nicholas Iuppa
Knowledge Industry Publications
701 Westchester Ave.
White Plains, NY 10604
$34.95

Educators Handbook for Interactive Videodisc
Ed Schwartz
Association for Educational Communications & Technology
1126 16th Street NW
Washington, DC 20036

Interactive Media
Diane Gayeski and David Williams
Prentice Hall
$12.95

Videodisc Book: A Guide and Directory
Rod Daynes et al
John Wiley and Sons
$89

Educational Technology
Educational Technology Publications
140 Sylvan Avenue
Englewood Cliffs, NJ 07632
$49/year

Optical Memory News
Rothchild Consultants
P.O. Box 14817
San Francisco, CA 94114-0817
$295/year

Video Computing
Videosoft
P.O. Box 3415
Indialantic, FL 32903
$130/year

Perspectives on Multimedia and the Future of Computers in Education: Dreaming and Planning the Future

Apple Computer has always been committed to listening to, and planning with, those in the educational community. Recently, Apple sponsored a series of "Education Connections" conferences across the country, during which the company shared some of its plans and hopes. Much of the discussion concerned multimedia (other topics touched on were connectivity and increasing access to information). The following was drawn from the opening address of those meetings.

Today, knowledge is fast becoming the new currency. With the incredible increase in the sheer amount of information to be mastered, students can no longer be passive absorbers—they need to become active thinkers and researchers. In shifting from passive to active, the entering freshman is doing activities parallel to the work of the researcher; in other words, the environments for instruction and research are beginning to converge. Many of those in the educational community are talking about how to use computers to enhance this new—more active—model of instruction.

Now, for the first time, tools are becoming available that will allow you to gather and access information, to manage and transform that information to create knowledge, and to effectively collaborate with others or communicate that knowledge in exciting new ways.

Apple Computer is hearing that educators want technology that is more powerful, easy to use, and accessible.

A number of needs relating to technology in education have been identified. Apple Computer is hearing that educators want technology that is more powerful, easy to use, and accessible—technology that will help you find new ways to make instruction and research both more effective and more engaging.

Apple is developing this technology in five key strategic areas. Technological breakthroughs in these areas will change the way we look at learning. They will change the way you access information, the way you transform information into knowledge, and the way you present this knowledge.

The five areas of greatest interest are as follows:

- **Multimedia Learning**—In day-to-day life, information comes in many forms; this should also be true of information available through computers. With multimedia, new models are provided for enriched learning, through integration of text, graphics, animation, music, speech, and full-motion video. These models allow for exploration through the connection of previously discrete or inaccessible information.

- **Publishing and Presentation**—Combinations of text, graphics, audio, and, most recently, video output, used to enhance the effectiveness of a wide variety of personal communications—written and verbal, to one person or a large group—in academic or administrative environments.

- **Curriculum Integration**—Enhancing current learning environments by integrating technology into the curriculum through access, ease of classroom use, and proper support. Providing new and exciting approaches to curriculum development, faculty training, and student learning.

- **The Scholar's Workstation**—A single computer offering a window to a world of diverse applications, and to information in remote locations and disparate formats. The scholar's workstation empowers students and faculty with new ways to explore, to create, and to learn.

- **Information Access and Integration**—Information should be accessible wherever it may reside, in the office, department, campus, or remote locations halfway around the world. Once this information is available, it should be easily integrated into classroom instruction, research, and administrative environments.

Let's examine these five areas individually.

Multimedia

Multimedia offers integrated learning by linking concepts through words, slides, lectures, video, animation, and simulation. This allows students to explore, link, and customize information on their own, while the professor focuses class time on helping them to interpret the data and make decisions.

The question then becomes, will integrated media truly be a better environment for instruction and research? We believe that the answer is yes. Here are two examples:

The question then becomes, will integrated media truly be a better environment for instruction and research?

- HyperPath, developed at Cornell University Medical College using a hypertext authoring tool called Guide, is a multimedia program that contains the entire pathology curriculum. This learning environment enables the student to review class notes, make direct links to scholarly journals and medical textbooks, access color graphics that depict complex physical organs and processes, or call up any of more than 24,000 videodisc images of x-rays or color slides. The technology enables students to review material in a meaningful nonlinear way and helps them cope with the explosion in medical information.

The experience at Cornell to date is that both faculty and students find HyperPath to be a more productive and stimulating environment than its noncomputer predecessor. Lectures have often become forums for discussion of the intuitive side of applying knowledge—instead of the high-speed data dumps they often were. In fact, the microbiology, physiology, and neuroscience departments are now joining the pathology department to create a rich web of medical education that links all four disciplines.

- At Brown University, the Institute for Research and Information Systems has developed a program called Intermedia, which is a multimedia research environment for scholars. Intermedia includes text and graphics editors, a "timeline" application, a relational database, and an animation editor, and has been used effectively in both English and biology.

Currently, Intermedia is being used in a 19th-century literature course. Imagine studying a work by Jane Austen. You decide to examine characteristics that the book's characters have in common with characters from her other novels. Through Intermedia, you can create links to these works, immediately reviewing characters and identifying common traits. Intermedia would also enable you to examine the art, music, and politics of the 19th century to gain an appreciation of the cultural influences of the time. This offers you unprecedented ability to customize, link, change, and evaluate information, to make discoveries, and to draw conclusions.

The true promise of multimedia lies in its ability to provoke insights.

Multimedia's offerings go far beyond the obvious benefit of enlivening the learning experience through vivid and engaging presentation. The true promise of multimedia lies in its ability to provoke insights by giving us a richly textured context that reveals relationships, dependencies, influences, and causes.

What's exciting is that materials for constructing rich multimedia environments already exist today. There are vast libraries of film—old and new, educational and entertaining. There are billions of feet of recorded sound information, from animal sounds to classical music. The challenge is to put into place the new social and economic structures needed to govern the ownership and distribution of multimedia.

Our direction for the future is to provide high-capacity read-and-write storage, so that you can have at your fingertips all the pictures of the National Gallery or all the symphonies of the great composers. We are working on networking and fiber-optic cabling that will allow you to move picures and sound rapidly across networks. And, of course, we are working on faster processing power and higher screen resolution, to enable the presentation of paintings and three-dimensional renderings accurate and detailed enough to be used for teaching and research.

Publishing and Presentations

The trends that are driving multimedia learning are also creating new opportunities for publishing and presentations.

Products like the Apple LaserWriter® printer have helped to produce documents such as grant proposals, research papers, campus newsletters, high-quality correspondence, and graphic art.

Consider the example of Dr. Lou Romano at Wayne State. As a chemistry professor, Romano must illustrate complex chemical formulae and detailed physical structures, often integrating these illustrations into text that requires special characters of its own. In the past, he'd have to take a graph to an artist, return two or three more times for editing and then pay $50 to $100. Now, with the LaserWriter, the Macintosh, and MacDraw software, he can produce a professional-quality graph himself in 10 minutes.

Today, technologies such as optical character recognition, audio digitizing, frame grabbing (taking a frame of video and converting it into a digital image), inexpensive color output, and video directly from the computer are leading to an expansion of our capabilities to create and communicate knowledge.

Today, technologies are leading to an expansion of our capabilities to create and communicate knowledge.

Imagine an environmental biologist who goes to a professional meeting to discuss the effects of different chemical emissions on the quality of air. Normally, she would take along her research in a basic text (and, perhaps, graphic) format. But in the future, we will see this same scholar simulating different combinations of chemicals in the atmosphere and making a video of the simulation directly from the computer. Text documentation will remain critical, but the verbal presentation of results would certainly be improved by a visual demonstration. Already, in fact, simulations are being done on supercomputers and displayed on workstations at places like the University of Illinois National Center for Supercomputing Applications. Moving the whole process to a workstation is not very far off.

The Scholar's Workstation

With the scholar's workstation, Apple will provide a tool that allows researchers to work in a way with which they are familiar. Yet it will also allow more efficient access to more information—optimally with more effective results.

Take, for example, Dr. Smith at the University of Indiana, a political science professor who is using Microsoft Word to develop a research article on "The Prediction of Voting Patterns for the 1988 Election by State." He displays his article in one window on his computer screen. In another window, he uses MapMaker to examine the statewide election results for the 1986 congressional election, and in a third window, he runs a STELLA model that

shows the underlying strategic relationships between how Democrats and Republicans vote on key issues. The scholar's workstation allows him to simultaneously run multiple applications, to access different information in various locations, and to link related information—in the process, developing a fresh perspective on projecting voting patterns.

Apple is working closely with the academic community to investigate new software technologies for the future.

The current directions of both the Macintosh operating system and the A/UX® environment show the potential of the scholar's workstation. In addition, Apple is working closely with the academic community to investigate new software technologies for the future—technologies that will allow you to go beyond graphics and text documents to integrating text and graphics with animation, video, and sound.

Information Access and Integration

Technologies are needed for accessing, searching through, classifying, linking, and managing information. For example, an "intelligent agent" could go out over the networks searching for information on the issue of deforestation; the system ferreting out data that seemed applicable, classifying it, correlating it to similar articles, and bringing it back to the user. Apple is striving to make that future a reality, to provide the ultimate window to the world of information.

Apple's strategy is to focus on innovative solutions that support important networking and communications standards in higher education, while providing new proprietary advantages where appropriate. This dual focus is represented by activities in several areas that result in the Macintosh being well-connected on campus—in labs, dormitories, and departments—as well as outside the campus. The following are the major communications systems in use today:

- The AppleTalk® Network System provides a low-cost, complete network architecture for classrooms, dormitories, and public access labs.
- DEC connectivity offers the friendly user interface of Macintosh, with the computational power, network architecture, massive storage, and tape backup capabilities of DEC systems.
- IBM connectivity provides 3270, 5250, and LU6.2 communications with IBM.
- TCP/IP for both the A/UX and the Macintosh operating systems allows Macintosh computers to communicate with a wide range of host computers both on and off campus.
- Finally OSI, a large-scale, industrywide standard to which Apple is committed.

Reed College in Oregon is using the Macintosh across campus, applying both innovative Apple technology and older communications standards. AppleShare® file servers are located throughout the campus on an Ethernet backbone. Instructors develop assignments and use these file servers to provide source material for research. Students have access to a number of computer labs around campus, where they can work on assignments and electronically send homework back to their instructors. This streamlines the distribution, development, and collection of assignments and allows class time to be used more effectively. Students and faculty can also access and search the library card catalog via the network. And even the campus newspaper, *The Quest*, has a "drop box" on the network for article submissions and letters to the editor, which simplifies the overall process for developing, collecting, and reviewing articles.

Apple's long-term objective is to bring the same kind of ease of use that we've brought to academic and productivity applications to networking and communications. To provide what we call "The Universal Access Paradigm." This means a powerful interface that not only lets you work with information in an effective manner, but also gives you transparent access to the sources of that information.

Apple's long-term objective is to bring the same kind of ease of use that we've brought to academic and productivity applications to networking and communications.

Looking into the future, a teacher might be sitting in front of his or her computer and have simple and consistent access to all of the following resources:

- a library card catalog for research
- an administrative database with student grades, transcripts, and attendance records
- a computer server for large computational problems
- a distributed file server (on which the teacher would have an assigned storage space)
- a wide range of applications
- a high-speed laser printer

In fact, at Reed and on many campuses, most of these services already exist. But the real breakthroughs will occur when the user doesn't have to care about the media or protocol being used; that is, when he or she is presented with *one* consistent way to send and receive information—in other words, when network services are taken for granted because they exist everywhere and function transparently.

It sounds wonderful to say that we now have access to the libraries of the world. But libraries hold information in forms other than text. Information comes in many forms—photographs, paintings, film, video, and sound, to name only a few.

In research, we traditionally go to libraries or do fieldwork and deal with various forms of media separately. In the future, historically different media have the potential to be completely integrated. The platform for integration is the personal computer.

In the future, historically different media have the potential to be completely integrated.

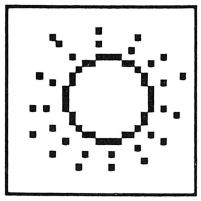
Implementing the Technology

Once we have all these great new technologies, how do we ensure that they are used in the academic environment?

Apple is committed to providing innovative programs from curriculum software solutions to software distribution, from financing to technical support—programs to effectively bring technology to your students, faculty, and staff.

As personal computers have become more available and powerful, we are able to convey more of the character and inner life that transforms information into knowledge. It is not just a matter of adding graphics or color or motion or sound. What's truly enriching our view of information is an emerging ability to capture, convey, and discover relationships. It is through a grasp of connections and interactions that true understanding and critical thinking develops.

Spotlight



The **Spotlight** section focuses on some of the Macintosh activities that are going on at various schools and universities. In this issue, it covers a diverse range of projects and programs—from the highly technical development efforts taking place at some schools to other schools' initial incorporation of computers into the instructional process.

University of Nebraska: AppleShare and the College of Business

Recent strides in personal computer connectivity have generated enormous interest in the Coe Computer Center and the College of Business at the UNL. Computers have become more and more a part of the college, but no specific system was designated, leaving the decision up to the individual user. As a result, the Coe Computer Center, a student lab, contains approximately 50 IBM PCs and 15 Macintosh computers, while individual departments within the college have a wide range of MS-DOS and Macintosh equipment.

It has now become necessary for faculty members and students to be able to share files between the different operating environments. Many different configurations were researched and evaluated. MacLink was used to transfer files between Macintosh and MS-DOS machines. This solved the problem of file transfer, but did nothing for file sharing. A small TOPS system was then implemented, but it was found to be too difficult to manage.

It has now become necessary for faculty members and students to be able to share files between the different operating environments.

We have recently found that AppleShare and MacLink provide the most successful solution. Coe Computer Center is implementing this project in three steps: 1) Introduce AppleShare in the student lab; 2) form a second network for the instructors using PhoneNET by Farallon, and link this network to the student's network using the Hayes InterBridge; 3) connect the networks to the campus mainframe backbone (HuskerNet).

The first phase has already been completed. The student network connects 15 Macintosh and two IBM computers, and one LaserWriter and two ImageWriter® II printers. The students adjusted to the change very easily, because the user interface remained the same in the transition from the disk system to the network system. The only difference is that the students no longer need to check out software, which is stored on the network. This was one of the most important factors in our decision.

The two IBM computers on the network are currently reserved for file transfer and laser printing. Most of the file transfers can be accomplished using specific software applications (for example, Microsoft Excel is capable of reading and writing Lotus 1-2-3 files, and Microsoft Word can read files created on MS-DOS and Macintosh machines). When this type of transfer is not possible, MacLink is used. This system has been very effective in solving many file compatibility problems.

The second phase, installing the instructor's network, is currently being implemented. This will involve the connection of computers belonging to some of the instructors in the college to a separate AppleShare network using PhoneNET (which allows an AppleTalk network to operate with existing phone lines). This will save time and money, by eliminating the need to string LocalTalk™ cabling throughout the building.

The second network will then be connected to the existing student network using a Hayes InterBridge.

The second network will then be connected to the existing student network using a Hayes InterBridge. This will allow faculty and students to share resources such as CD-ROM drives, LaserWriter printers, and mainframe access paths.

A second reason for this link is to form an electronic communications system between the students and faculty. Students will turn in assignments on the network instead of on paper. The assignments will be corrected and returned to the students using the network. Although AppleShare allows for this type of file transfer, several electronic-mail programs are currently being evaluated.

The third and final stage will be the connecting of Coe Computer Center's AppleShare network to HuskerNet, the school's mainframe backbone. The connection will be made in the instructors' network, but will be accessible to the students through the InterBridge system.

For further information, contact:
Martin Billington
240 CBA, University of Nebraska—Lincoln
Lincoln, NE 68588-0405
(402) 472-5712
AppleLink: ST0017

Drexel's Co-op Disk: Preparing for a Job Electronically

By the time they graduate, most Drexel University students really know how to work. That's because 95 percent of the university's undergraduate students participate in the Co-op work program, finding jobs through the Center for Cooperative Education and Placement (Co-op Office). Faced with the costly problem of producing brochures yearly, with text and graphics that were expensive to update, the Co-op Office turned to the electronic capabilities of the Macintosh, asking the Office of Computing Services (OCS) to create the Co-op Disk.

The Co-op Disk gave the Co-op Office the financial flexibility and ease of updating it needed to communicate policy changes, new academic requirements, and general information to new students in the program. The disk replaced the brochure and simultaneously created a learning tool for freshmen and transfer students attending Career Planning and Professional Development, a required academic course for Co-op. Among the Co-op Disk's many features are sample resumes and guidelines for writing resumes, two quizzes on corporate etiquette, a GPA calculator, an interactive tool to prioritize career choice (or any other) criteria, and copies of the forms that students must fill out before they begin their Co-op experience. Future enhancements to the program include the ability to fill out and print these forms (or download the completed forms into a database on the Co-op Office's hard drive), and the ability to communicate new information to students within 24 hours, by placing the disk on a file server in the central OCS computing facility. Extra or outdated Co-op Disks can be recycled when a new version is released—which is considerably cheaper than throwing out old brochures and paying for new typesetting and design services.

Among the Co-op Disk's many features are sample resumes and guidelines for writing resumes.

The Co-op Disk was designed by OCS and modeled after the Drexel Disk, a disk given to each entering freshman and transfer student that contains graphics and information about OCS and Drexel in general. The Co-op Disk and Drexel Disk were developed in Pascal by programmers in the OCS Software Development Group, under the guidance of a Drexel faculty member in the Department of Psychology and Sociology.

For more information, contact:
Dr. Jan Biros
Coordinator of the Microcomputing Program
Drexel University
(215) 895-2667

Drexel's Publication Development Center: The Art of Technological Literacy

Students with a modem at home can send Macintosh word processing files to the typesetter in the Center.

The Publication Development Center in Drexel University's College of Design Arts is bringing technological literacy to various arts groups. For three years, the Center has been the training ground for future graphic designers, publications managers, and print technology managers. One of the required courses for students in these three academic programs, Computer Image Generation and Telecommunication, utilizes the Center's five Macintosh SE computers networked with AppleTalk to two LaserWriter Plus printers, one Compugraphic typesetter, one Compugraphic scanner, and six Compugraphic terminals. Telecommunications packages make it possible to interface the two systems. (Red Ryder is the preferred package.) Students with a modem at home can send Macintosh word processing files to the typesetter in the Center.

The center is even bringing technological literacy to children, through a new publication called *TIES (Technology, Innovation, and Entrepeneurship for Students). TIES* is a cooperative, nonprofit publication for K–12 educators interested in helping their students to increase their technological literacy and capability. It is provided at no charge to technology teachers. Dr. J. Michael Adams, Dean of the College of Design Arts, explained the mission of *TIES* in its premiere issue: "One goal of *TIES* is to foster that special quality of informed originality that stems from the acquisition of design/problem-solving skills. Students with these learned skills will make better employees, more effective managers, and we believe, will become more critical thinkers in all areas of their lives. The second goal is to provide a medium where knowledge is omnidirectional, where technological innovations are made comprehensible to students and teachers, and where they in turn can use their knowledge to have a positive impact on technology."

The Publication Development Center was created to provide Drexel faculty with an inexpensive means to develop scholarly publications and provide hands-on experience for master's students in the Publications Management and bachelor's students in the Graphics Design and Print Technology programs. The faculty and students at the center design and typeset *TIES, Neuropsychology Journal, PSLS (Publication for the Society of Literature), Career Forum* (for the College of Business and Administration), and many other publications for the College of Design Arts.

For more information, contact:
Peter Bartscherer
Publication Development Center Director
Drexel University
(215) 895-1636

Microcomputers in Rehabilitation: A Special Place for HyperCard

The past few years have seen an exponential increase in the use of microcomputers in general rehabilitation. Most applications and some creative input devices have been designed for people with sensory or motor handicaps who are cognitively competent. On the other hand, the microcomputer could also be an aid for the cognitively impaired, including some elderly people or those who have suffered a traumatic brain injury. In the past, the microcomputer has typically been used only occasionally as a tutor, with the hope of providing general training in cognitive capabilities like attention, concentration, and spatial and temporal relationships. At best, outcome studies show marginal results, particularly regarding generalizability to "ecologically valid" daily living activities. Software was often originally designed for K–12 education or as games that were employed in this tutoring role.

The microcomputer could be an aid for the cognitively impaired.

The neuropsychology program at Drexel University and the Drucker Brain Injury Center at Moss Rehabilitation Hospital in Philadelphia have been developing an alternative approach, which uses the computer not as a tutor but rather as a tool. The computer performs prosthetic functions individualized to the "ecological" requirements of each patient. For example, a grant from the Solon Foundation to Dr. Douglas L. Chute has provided for basic research on interface design for the cognitively impaired and for applied research on prosthetic applications for the traumatically brain-injured patient. The programming resources of Drexel's Office of Computing Services, in the Software Development Group, have produced related instructional and courseware materials such as the MacLaboratory for Psychology series.

Use of the microcomputer as a cognitive prosthesis presents the problem of requiring unique software design and development. Until recently, this precluded anything but a "cottage-industry" approach. Now, however, powerful, professional, and readily customized "shell" programs such as HyperCard and interfaces such as MacLaboratory for Psychology Controller are available. These require little if any computer programming to produce important prosthetic devices for the cognitively impaired. Although our work at Drexel is preliminary, we can conclude that the microcomputer as a "tool" is an important new aid supporting the daily living activities of individuals with cognitive difficulties. At this time,

Use of the microcomputer as a cognitive prosthesis presents the problem of requiring unique software design and development.

beta versions of ProsthesisWare suitable for expressive aphasics, SpeakEasy and SpeakEasier, are available.

For more information, contact:
Dr. Douglas Chute
Neuropsychology Program Director
(215) 895-1722
or
Dr. Melanie Hoag
Software Development Group
(215) 895-1876

University of Wyoming: Demonstration of Fundamental Kinematic Systems

These programs have as their objective an enhanced student understanding of the relative motions that occur in fundamental kinematic systems.

Over the past few years, a series of programs has evolved through a cooperative effort between the University of Wyoming and Ocean County College (N.J.). These programs have as their objective an enhanced student understanding of the relative motions that occur in fundamental kinematic systems. The initial programs included simulations of the basic kinematic chains—the four bar linkage, the slider crank mechanism, and the inverted slider crank. These programs were supplemented with others that simulate two loop machinery systems, where each of the two loop systems is driven by one of the fundamental kinematic devices. All of the programs have been incorporated into a single program with pull-down menus to allow selection of mechanism type.

The program includes considerable user interaction; to the point where a user can define a particular configuration on the screen and observe the resultant motion of the system. The speed of the display is controlled—within reasonable limits—by the user, and the perception of continuous motion is quite good.

For more information, contact:
Dr. Donald A. Smith
Department of Mechanical Engineering
University of Wyoming
Laramie, WY 82070
(307) 766-6495

If you would like a copy of the program, please send a blank initialized disk.

Hamilton College: A Project to Improve the Reading/Communication Process

Developing effective communication skills, including the ability to write with precision and cogency, is an essential part of the mission of an undergraduate education. Much has been said in recent years about the ways in which computing can assist with the writing process, spurred on by the widespread use of the microcomputer in higher education. Software has been developed to assist writers in making the connection between thought and the printed word (for example, outlining/prewriting programs, syntax and spelling checkers, and content evaluators).

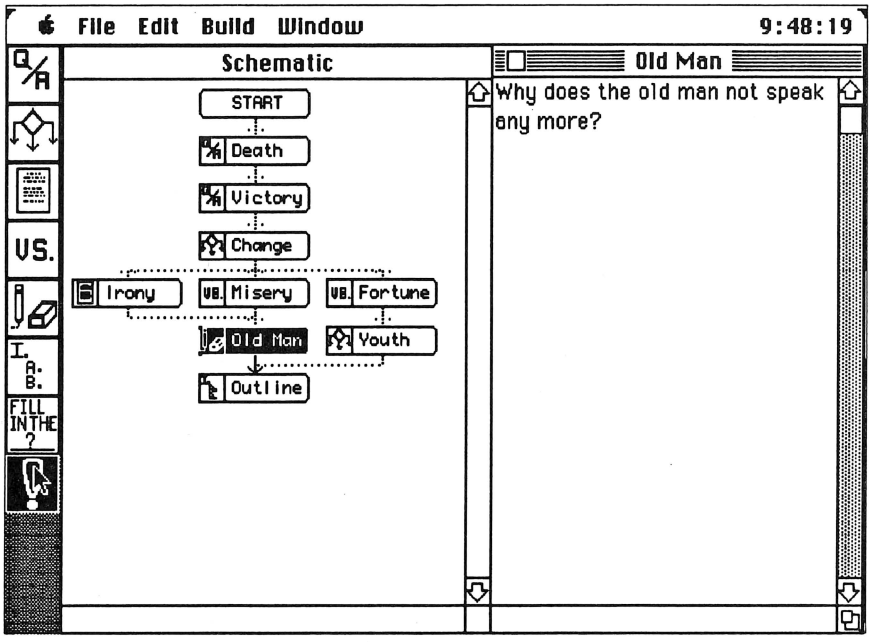
Hamilton's efforts are based on the idea that reading carefully and talking sensibly about what one has read are at the center of a liberal arts education. Effective reading skills are an essential precursor to meaningful classroom discussions, and such discussions contribute to the development of ideas necessary for effective writing. In particular, we seek to develop ways to use computer-aided instruction to help students be better prepared for classroom discussions. The best discussions, we believe, are those in which students try out positions and ideas that can be questioned and supported both by fellow students and by the instructor. The development of effective reading skills is especially important in the liberal arts environment, which depends heavily on class discussion and the writing of essays and theses based on close attention to poems, novels, philosophical and political treatises, and the like.

Hamilton's efforts are based on the idea that reading carefully and talking sensibly about what one has read are at the center of a liberal arts education.

The results of inadequately engaging assigned material are familiar to every faculty member—disappointing class discussion, superficial understanding of the assigned material, and lack of development of the ideas that would form the grist of written assignments. Students may turn and mark the pages and even take notes, but increasing numbers do not engage the themes or arguments of the text. The skills necessary to do so must be developed and exercised; they are not innate.

The Hamilton project addresses this issue. Titled "Technology in the service of the liberal arts," it is partially funded by a grant from the Pew Memorial Trust. Software called DISC-US (pronounced "discuss") is being written to encourage the development of critical reading and thinking skills. The goals are straightforward: to improve the quality of student reading and critical-thinking skills, of classroom discussion and written expression, and, ultimately, of students' overall performance.

The goals are straightforward: to improve the quality of student reading and critical-thinking skills, of classroom discussion and written expression, and, ultimately, of students' overall performance.



DISC-US is built around the simple Macintosh interface. It has been designed with the intention of requiring, encouraging, and facilitating student interaction with texts and with each other in the discussion of texts. It comprises three separate systems: a MacPaint-like authoring system (see the above figure), that provides teachers with a palette of highly general question types for building lesson schematics, and allows them to create, edit, and test lessons interactively, and to link them with course-related information; the student portion of the program, which "runs" lessons and solicits and saves student responses on a centralized database (access to the database over the network can be specified by the author of the lesson to permit students to see their classmates' responses to questions and to review their own previous responses); and a utility program that allows instructors to examine the database, monitor student progress, and even send messages to individual students or to classes as a whole.

The hardware environment is a local area network of Macintosh computers with file and print servers. In addition, the local network will be accessible from other local networks and stand-alone machines via our recently installed Rolm CBX telephone switch.

Our goal is, in part, to create an easily learned authoring system for instructors that will enable them to create "lessons" on reading assignments, lessons that will help students develop ideas to test. These lessons will not be drills in which the correctness of a student's responses is evaluated against a model held within a computer program, nor will they simply offer unstructured time for student-to-student interaction on the network. We envision lessons that require students to make interpretive choices, and then present them with the consequences of those choices. Lessons can branch into different paths depending on how students respond to certain

key questions. Students may also be asked to respond along the network to the arguments of their classmates, so that they can begin to appreciate the multiple perspectives possible on some important issues.

We intend that the authoring system be flexible enough to accommodate a wide variety of critical methods and reading assignments.

A multidisciplinary team of faculty from the departments of political science, English, and computer science, as well as teams of undergraduates, developed a prototype system during the first six months of the project. Further, a special center was constructed to house the equipment and serve as an experimental laboratory for project activities. A conference to share preliminary results of the program will be held this summer.

Advanced technology can be used to improve students' ability to decode and engage texts. We are also convinced that the development of this skill in students can do more than virtually anything else to improve the quality of formal discussion and the general educational climate at institutions of higher education.

Finally, the disciplines of music, foreign languages, and speech are important participants in the "Technology in the service of the liberal arts" project. The experience gained in developing programs for understanding and discussing texts will also be applied to learning the nonverbal "languages" of music and visual art. The overall impact of this project on the quality of the undergraduate experience can be significant.

The overall impact of this project on the quality of the undergraduate experience can be significant.

The DISC-US system was used in the fall semester in two limited-enrollment sections—Introduction to Political Theory and Introduction to Literature in English. These courses provided a variety of texts, from sonnets to plays to longer treatises in political theory, for experimenting with DISC-US lessons. Students in each course were required to do weekly lessons that were the basis for class discussion. The lessons were designed to use all the features of the authoring system and the AppleTalk network.

The results of this initial use are promising. Students were generally positive about the assigned lessons, comparing them to laboratory work in the physical sciences. The two instructors found that the students were better prepared for class discussion and better able to refer to the text in speaking to points under discussion. On the negative side, it was difficult to go very far through "branching" in order to draw out lines of argument, and the network did not provide as much useful student dialog as might have been desired. These difficulties, however, are less indicative of problems with the system than of the need to develop imaginative ways to utilize it to its full capacity.

We will be exploring the possible uses of DISC-US in the same two courses this spring and through a conference this summer (July 13–16).

For further information about the system or the conference, contact:
David Paris
Department of Government
Hamilton College
Clinton, NY 13323
(315) 859-4224

San Jose State University: Spartan Bookstore Macintosh Training Lab

By Julie Arimura, Spartan Bookstore Computer, Electronics Dept. Apple Student Representative

After selling more than 2,000 Macintosh computers, Spartan Bookstore at San Jose State University recognized the need for hands-on Macintosh training in the SJSU community. Last summer, the bookstore fulfilled that need by opening its own Macintosh Training Lab.

Macintosh labs on college campuses are becoming more common, but a Macintosh lab in a college bookstore is still unusual.

Macintosh labs on college campuses are becoming more common, but a Macintosh lab in a college bookstore is still unusual. Just beyond the sales floor, a former stockroom was transformed into a training lab. Three simulated granite tables support nine dual-floppy-drive Macintosh SE systems, each with an external 30-megabyte SCSI hard disk. The two video screens on adjacent walls are connected to the video Macintosh that rests upon the instructor's desk at the head of the room; students can see exactly what the instructor is doing at the Macintosh. One wall is lined with bookcases filled with software documentation and tutorial materials including audio- and videocassettes.

Spartan Bookstore training lab users range from new Macintosh owners receiving "Introduction to Macintosh" instruction to those with more specialized needs such as word processing, graphics, and spreadsheet/database management. ("Introduction to Macintosh" classes are free; other classes require a nominal fee.) The lab is also used by its staff, who receive training from Apple and third-party software trainers.

Student employees handle the actual training. The students, sensitive to the academic needs of San Jose State, prepare much of the training curriculum.

For further information, contact:
Kevin Warner
Computer/Electronics Manager
Spartan Bookstore
San Jose State University
San Jose, CA 95192
(408) 924-1806

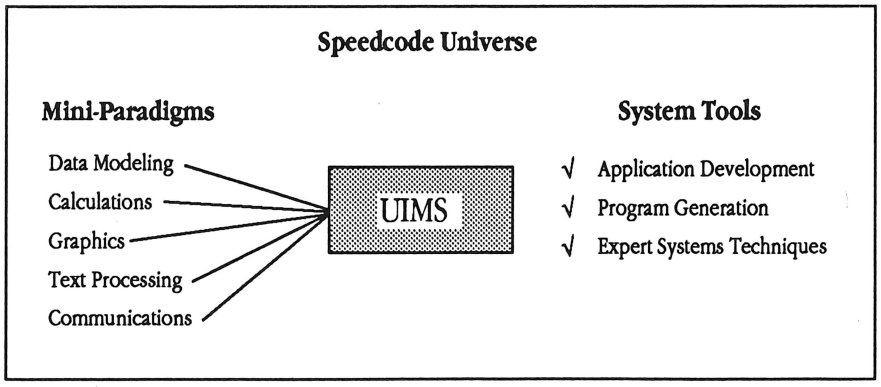
Oregon State University: Speedcode Universe

What Is OSU?

OSU (Oregon Speedcode Universe) is an experimental programming system designed for Macintosh programmers who want to produce applications very rapidly. The aim of OSU is to increase programmer productivity by as much as a thousandfold, by combining rapid prototyping, reusable units, program generation, and expert-systems technology into an integrated environment that we call "speedcode universe."

The aim of OSU is to increase programmer productivity by as much as a thousandfold.

The diagram shows the system's overall structure. The heart of OSU is called the UIMS (User Interface Management System). The UIMS allows a programmer to create and directly manipulate icons, menus, windows, dialogs, alerts, and user-defined procedures. It consists of three parts: 1) a resource editor called RezDez (Resource Designer) for WYSIWYG (what you see is what you get) creation and editing of icons, menus, windows, dialogs, and alerts; 2) a WYSIWYG "sequencer" for "training" the application to behave the way it should when in actual use; and 3) a program generator that writes a Pascal source code program that implements the behavior specified by "sequencing."



Clearly, different kinds of applications require different kinds of tools; word processing is radically different from graphics processing. To achieve extremely high levels of productivity without sacrificing code quality or flexibility, OSU relies on a collection of "domain-specific" tools called "mini-paradigm" editors. A mini-paradigm for designing database applications, for example, might be an Entity-Category-Relation diagramming editor for defining an application's data. We call this the Data Model Paradigm. Similarly, tools for graphically defining calculations, animation, and data communications/networking must rely on different ways of viewing the application. Several mini-paradigm tools are currently under investigation and development, and will be described in later reports, but this discussion is limited to describing the UIMS.

The results of using OSU can be conceptualized in a number of ways, depending on how the system is used. You might think of OSU as an *application development system* if it is used with the Data Modeler or Calculations mini-paradigm to produce database applications or mathematical models. Alternately, you might classify OSU as a *program generator* when using it to automatically write a program that manages the user interface for your application. Further still, you might consider OSU to be an *expert systems shell for deriving complete applications from rapid prototypes,* if you use its facilities for building an application from reusable components, user requirements, and programmer-supplied rules of behavior. OSU is all of these rolled into a unified environment for radically increasing programmer productivity.

OSU is currently experimental and incomplete.

OSU is currently experimental and incomplete. We do not yet know how successful it will be in terms of increasing programming speed. At this point, we can prototype all standard user-interface objects defined by the Macintosh Toolbox. As a result, most of the programming required in a typical application can be done in a day or two. Thus, tenfold improvements on the speed of generating "standard applications" are within reach. Our goal, however, is much more ambitious—consequently, OSU will not actually not be completed for two or three more years. In the remainder of this report, I will describe how a programmer uses OSU to construct an application program without using a programming language.

How Is OSU Used?

A programmer uses OSU to design and implement an application program by drawing the user- interface objects on the screen, using a graphical resource editor similar to other resource editors available for the Macintosh. The graphical objects are written to the application's resource fork and then used by the programmer, who "simulates" the operation of the application by "sequencing" the user-interface objects through direct manipulation.

For example, consider a simple application consisting of a File menu and a Calculation menu. Further assume that each menu contains two items: File-Open, File-Quit, Calculation-Solve, and Calculation-Show. The purpose of the File menu is obvious; the purpose of the Calculation menu is to solve some mathematical problem and to show the results as a line graph.

The programmer "trains" the application by actually running it.

OSU is used to design the desktop icon for this application, enter the two menus and their two items, and create all dialogs, windows, and alerts associated with running the application. The programmer then "trains" the application by actually running it—more specifically, the programmer carries out the sequence of operations that the application is supposed to do.

For example, when the application is launched, suppose a "welcome" dialog is supposed to appear on the screen. The programmer creates this dialog by drawing it on the screen and saving it in the application's resource fork. Next, suppose the application activates a sequence of user interactions when the Calculation-Solve menu item is selected. The sequence might contain a dialog for collecting the input values from the user, an error alert used in case the user enters improper values, and a result dialog containing the numerical results of the calculations. All possible sequences of events that may occur when the user selects Calculation-Solve are "prototyped" by the programmer.

In the scenario above, the sequences of user interactions were easily supplied by simply doing what a user might do. The sequence is "vacuous," in the sense that the actual calculations are missing. This is where the programmer must resort to a mini-paradigm. A Calculation mini-paradigm is used to gather the information needed to do the intended calculations; a "plan" is produced from the Calculation mini-paradigm and is "plugged" into the sequence at the correct place. A "procedure" object is defined in the sequence as a "placeholder" for plans. When the programmer wants to add the plan to the sequence, he or she does so by inserting a "procedure icon" into the sequence—after the data input dialog disappears and before the result dialog appears.

A graphical display of the result is created in a similar fashion—the Graphics mini-paradigm editor is used to produce a "plan" for graphical display, and this plan is inserted into the sequence at the proper place.

The completed application prototype can be "played back" to verify the correctness of the program. When played back, the prototype behaves as instructed by the programmer. No calculations are done; only the sequence of operations is repeated.

The application prototype is converted into a stand-alone application by a built-in program generator, which writes a Pascal source code equivalent of the prototype. During this phase, the "plans" that were created by various mini-paradigm editors are written as source code modules. These modules perform the desired operations, thus realizing a complete application.

Research at Oregon State University

OSU is an example of the work going on in the software engineering research laboratory, sponsored by the computer science department. It was developed over a two-year period by Jim Armstrong, Sharada Bose, Fred Handloser III, Sherry Yang, and others. Work on OSU continues, especially in the area of inventing and implementing various mini-paradigms for generating plans.

We are also working on model management systems for nonprogrammers. For example, we are developing a linear programming system for devising very large linear models and the databases they require. Some of these models are so large that they require special-purpose parallel processors.

A VHS videotape demonstration of these ongoing systems development projects can be obtained by sending a blank cassette to the address below:
Software Engineering Lab
Computer Science Department
Oregon State University
Corvallis, OR 97331
Attn: T. G. Lewis
(503) 754-3273

Ohio State: An Engineering Approach to Software Design

The Reusable Software Research Group at The Ohio State University is examining a wide range of technical problems related to reusable software modules, including specification and verification, efficiency of implementation, programming language issues, and many others. An important part of our work is the development of a programming environment called STILE (STructure Interconnection Language and Environment) that will enable us to explore a particularly promising approach to reusable software—the "engineering design metaphor." The idea is to view software parts much as reusable physical parts—such as gears, integrated circuits, and valves—are viewed by the traditional engineering disciplines.

MacSTILE is the system we have constructed to support this view of software design and development. It is, in effect, a CAD/CAM system for software and is, therefore, superficially similar to existing CASE systems and to schematic capture systems for electronic circuit design. However, MacSTILE differs from such tools in several important ways, providing better methodology

support and a more consistent human interface. We hope to demonstrate how these features can lead to superior software designs and improved programmer productivity.

In MacSTILE, a system consists of *parts* created by using an interactive graphical editor that enforces a consistent syntax on graphical descriptions. This graphical editor is syntax-sensitive, but knows nothing about the semantics (meanings) of parts and is, in this sense, domain independent. Its use, therefore, is not limited to software system design; it can be used in any situation where hierarchical block diagrams or schematics might be drawn to describe a "system." Each part's inherent meaning is determined by its *interpretation*, which is a context for understanding that part's function. A file containing a description of the parts and how they have been composed is produced as the output of MacSTILE and can be used as the input to a postprocessing program for that interpretation. This postprocessor mechanically supplies the meaning of a system schematic from the meanings of its component parts and the interconnections among them. One such interpretation, for example, considers a MacSTILE schematic as a description of a set of communicating software processes; in which case, the postprocessor might generate executable code for those processes. Another interpretation involves queues, servers, and their interconnections; its postprocessor might generate a simulation of a queuing network model described with MacSTILE.

Our objective for the near term (1988) is to demonstrate how MacSTILE can be used effectively in three apparently quite different application domains: to describe real-time systems and their associated control software; to describe image-processing "pipelines" for computer enhancement of radiological images; and to describe music synthesis. The work on image processing is being conducted by colleagues at the Image Analysis Research Center of the Medical College of Ohio in Toledo, while the other projects are being done at OSU.

Currently, we are also working on a companion tool that uses the engineering design metaphor as the basis for administration of a multiperson design effort. It provides the usual features of a version control and configuration management system in an especially straightforward way, along with several important new features. Our goal here is to evaluate this system in conjunction with MacSTILE in an experimental section of an existing undergraduate software group project course at OSU during the 1988–89 academic year.

The importance of improved programmer productivity and improved software quality is widely known in the computer science community. Computers should be expected to assist in the task of software design and development much as they do now in design of mechanical and electrical systems. MacSTILE is one effort to make computer-aided software engineering a reality.

Students at other schools can easily take advantage of the flexibility of MacSTILE.

However, as noted above, the potential benefits of MacSTILE are not limited to software engineers. Some OSU students are currently using MacSTILE for design projects both in software and in other domains that do not involve software as the end product. They are learning how computer-assisted design can be applied in a wide range of application domains, which gives them a better understanding of the larger arena in which design methodologies and computer-based support tools can be applied effectively. For example, individual students are writing postprocessors that interpret parts as denoting executable software modules (for example, real-time control software), that drive attached hardware (for example, a music synthesis system), and that simulate various system models (for example, general continuous nonlinear systems). Students at other schools can easily take advantage of the flexibility of MacSTILE and use their own imagination to apply it to application domains we had not even thought of when we built it.

We knew from the beginning, over two years ago, that the kind of environment we envisioned would require a sophisticated workstation with high-resolution bit-mapped graphics and a pointing device. We designed and started development of STILE for a UNIX workstation, but it soon became clear that the Macintosh would be a better platform for our work. Our frustration peaked just about the time we had finished writing our own scroll bars, dialog boxes, and other user-interface components for the UNIX workstation—a nontrivial task given the very low-level routines available at that time—when the vendor replaced the entire set of low-level software with a new and incompatible version. We immediately moved our development effort to the Macintosh using Lightspeed C, which has been an incredibly more effective software development and target system than the UNIX workstation.

We are anxious to demonstrate the effectiveness of MacSTILE in other disciplines as well, and at as many universities as possible.

Furthermore, because Macintosh computers are popular throughout all kinds of departments at most universities of all sizes, we believe MacSTILE can be more accessible to people outside of the computer science field than would a comparable system running on a UNIX workstation. We are anxious to demonstrate the effectiveness of MacSTILE in other disciplines as well, and at as many universities as possible.

We will, of course, use MacSTILE and (for next year's experimental group project course) the companion MacSTILE System Administrator tool now under development. The output file from MacSTILE that is input by a postprocessing program is a text file whose format is provided as part of the documentation, so any development system can be used to write a postprocessor.

The Computer and Information Science Department at Ohio State has more than 300 Macintosh Plus computers, all of which are networked by AppleTalk in small groups of up to 20, and which are connected by Kinetics gateways to the CIS department's Ethernet backbone, and, through it, to the campuswide network (SONNET, which is built on a Pronet-80 fiber-optic network). File service is provided by Pyramid computers. Of course, there are many other Macintosh computers around the OSU campus, in department, faculty, and staff offices, and in student labs.

There are many Macintosh computers around the OSU campus, in department, faculty, and staff offices, and in student labs.

The CIS systems are used primarily for introductory computer science instruction, but are also popular among faculty, staff, and graduate students, who use them for a variety of document processing, drawing, and database applications. We will use them in the experimental course with MacSTILE next year.

Our other computer science instruction is being shifted from time-sharing to a large network of UNIX workstations. Implementation of this plan is about 50 percent completed; by fall 1988, the CIS department will have moved completely to UNIX workstations (mostly Suns) for instruction beyond the introductory courses. By that time, we will have about 250 UNIX workstations to complement the Macintosh computers and a variety of other research and instructional equipment.

For further information, contact:
Professor Bruce W. Weide
Computer and Information Science
The Ohio State University
2036 Neil Ave. Mall
Columbus, OH 43210
(614) 292-1517
weide@cis.ohio-state.edu

California State University: Macintosh and Student Organizations

The Associated Business Students Organizations Council (ABSOC), which represents 10 professional and departmental organizations at California State University, Long Beach, needed to reach more than 7,000 business students to get them involved in the School of Business Administration. An IBM PC using MS-DOS provided inadequate data storage. In addition, many of the student leaders were unfamiliar with microcomputer usage.

With the encouragement and support of Higher Education Account Executive Bob DeWitt and Apple Student Representative Dan Gray, ABSOC began using a Macintosh SE Hard Disk 20 system and a LaserWriter Plus printer in its club office early this spring semester.

ABSOC's goal is to set a precedent for the use of computers within student-based organizations.

ABSOC's goal, as the student council representing the greatest number of students in the university, is to set a precedent for the use of computers within student-based organizations and to establish an environment that will be used as a model for other student councils at CSULB to follow.

We have been educating the leaders of our student organizations through weekly workshops put on by our Apple Student Representative. Topics have included word processing, desktop publishing, and database management. We are also successfully teaching these students to use the Macintosh and LaserWriter in our everyday work.

Presently, over four separate newsletters are published each month by our clubs and at least ten organizational flyers are generated each week. Five out of our ten groups have already set up database systems on the microcomputer, and we plan to have all of the organizations on-line by summer.

For further information, contact:
Paul Brennan
ABSOC Chairman
California State University, Long Beach
1250 Bellflower Blvd.
Long Beach, CA 90840
(213) 498-5370
BITNET: JBCK700@CALSTATE

Notre Dame: Computing Center Macintosh Lab

The Notre Dame Computing Center operates a public-access Macintosh lab that is being used as a pilot for setting up workstation clusters around the rest of the campus.

The Computing Center's Mac Lab currently supports 33 Macintosh Plus and Macintosh SE workstations. In addition, the User Services staff uses the same network running a variety of Macintosh Plus, Macintosh SE, and Macintosh II workstations. There are three printers on the network—two LaserWriter Plus laser printers and an AppleTalk ImageWriter II printer. One LaserWriter is available through the LaserShare™ print spooler, the other is available directly over the network, as is the ImageWriter II. Laser printing is free and total laser output is more than 100,000 pages per semester. The lab is served by a Macintosh II that acts as both a filer server and a print server. It is currently configured with 2

megabytes of RAM and an 80-megabyte internal hard disk running AppleShare and LaserShare™ network software. An Apple Tape Backup 40SC system is used for monthly server backups. The lab is open and available 24 hours a day, 7 days a week; there are approximately 50 active nodes on the network at any one time.

The Mac Lab is right next to the mainframe computer operations room. When consulting staff is not on duty, the operations staff of the Computing Center monitors the lab, keeps the LaserWriter printers stocked with paper, and notifies User Services staff in the event of an emergency.

A wide variety of applications are available for use on the network. Word processing, graphics/CAD, desktop publishing, spreadsheet, database, statistical, and project-management software, as well as programming languages and academic courseware, can all be accessed from different folders on the file server. Disk space has also been made available to MadMacs, the Notre Dame Macintosh user group, which maintains a large library of public domain software on the server. A separate folder of HyperCard stacks is maintained by the Computing Center staff.

There has been a lot of very positive feedback from faculty, staff, and students about the network. AppleShare is completely transparent to the user—each workstation automatically logs onto the network and the 40-megabyte server acts just like a Macintosh external drive. With a minimum of training, faculty, staff, and students can very effectively use resources available on the network. The only complaint has been that there are not enough Macintosh computers available. Typically, all machines in the Lab are in use, and, near the end of the semester, there is a waiting list 24 hours a day.

The only complaint has been that there are not enough Macintosh computers available.

Notre Dame is in the process of implementing a campuswide network. Plans for the lab's AppleShare network include an Ethernet gateway to this campuswide network and access to the campus mainframe environment. We are also planning for a high-end workstation lab, and looking forward to using the Macintosh II/Ethernet environment.

For more information, contact:

Scott Cowdrey
Assistant Director for User Services
Computing Center
University of Notre Dame
Notre Dame, IN 46556
(219) 239-5600
BITNET: C3WRFK@IRISHMVS

Joel Cooper
Consultant/Analyst
User Services
Computing Center
University of Notre Dame
Notre Dame, IN 46556
(219) 239-5600
BITNET: CPCTAJ@IRISHMVS

Gonzaga University: Attention Schools of Education

The School of Education saw a need to educate itself and its students in terms of computer abilities for professional growth and development.

This article grows out of the need to share our experiences with other Schools or Departments of Education across the United States that may face the same problems we did. Three factors moved us toward computerization. First, there is the increase in the number of enrollees who are seeking teacher certification or advanced degrees in a specialized area of education. Second, our School of Education is not fortunate enough to be located in one contiguous space: it is spread out over four floors in one building and in another building across campus. And third, the School of Education saw a need to educate itself and its students in terms of computer abilities for professional growth and development.

Consequently, after two year of surveys and studies, the School of Education at Gonzaga University decided not only that computers were necessary to help manage the school, but also that a network was required to tie faculty, staff, and administration together into a cohesive and functioning group not separated by physical boundaries.

Our computerization started a year ago with the Dean's office complex, where four Macintosh SE computers were installed and linked via AppleTalk to a LaserWriter Plus printer. The benefits of ease of use and quick networking were realized immediately. There were some problems with incompatible Finder versions and application software, but, on the whole, this first installation was successful. Other faculty members watched carefully how the system worked, and noticed the benefits it offered in terms of efficiency and improved work quality.

Through the AppleTalk network, the Dean, her assistant, and her administrative assistant can share appointment calendars, retrieving last-minute, up-to-date information on meetings and conferences. One calendar was established to monitor conference room activities, so everyone on the network could see when the conference room was available. Because this office monitored and maintained the financial activity of the school, staff would typically print pages of numbers, and wait for approval—only to find out that changes were necessary, and the process must be repeated. But with AppleTalk and desktop communications, a Microsoft Excel spreadsheet draft budget could be sent via InBox; the budget didn't have to be printed until it was finalized. Perhaps the most important feature of the system was the ability to send and receive communications electronically. All of the network users agree that this software not only saves time and energy, but also got rid of all the little pink slips of paper that had been floating around on their desks.

The next phase of computerization was to reach outside of the Dean's office complex and bring other departments onto the network. Administration, Curriculum, and Instruction decided to buy four Macintosh SE systems and add them to the Dean's complex. This time, there were no problems with hardware and software. The drawback was that the LocalTalk cabling could handle only short distances, and the departments involved were four floors apart. Fortunately, PhoneNET came to the rescue. After much time spent with the phone company, our Macintosh network immediately expanded to seven computers and a LaserWriter printer.

One year later, we are 10 Macintosh systems, one LaserWriter Plus printer, and two Shiva NetModems strong. There are plans for adding three IBM compatibles via LocalTalk PC, as well as more Macintosh computers. It is hoped that we will soon have access to the university's campus network.

Outside of our campus, the School of Education serves 14 centers throughout the Pacific Northwest and Canada. Our current plan is to establish a computer bridge between each center and the computer center within the School of Education. The goal is to link off-campus students with the main campus, allowing remote CD-ROM database access for professional research.

Our web of communication is growing. People are beginning to think that exchanging information with their computers is possible, and even more important, not difficult. Questions like, "When will I be able to do my paperwork at home with my computer, and come to school to teach and really spend time with colleagues and students?" or "When will I be able to register for courses at Gonzaga from my computer at home (school)?" are becoming more common. Programs such as InBox, 4th Dimension, Microsoft Excel and Word, Smart Alarms, TOPS, and PageMaker have allowed us to extend our professional functionality and enhance our personal creativity.

People are beginning to think that exchanging information with their computers is possible, and even more important, not difficult.

So far, the School of Education has moved far ahead of any other department in the university in terms of computerizing its functions. Because of the network's popularity and ease of use, we have already reached our limit in terms of adding any more terminals: future additions will require us to boost and repeat the network signal. The increased number of networkable nodes in each department now necessitates segmenting the network into zones, either by floor or by department.

I hope this article will help other Schools of Education with the physical dilemmas they may face in instituting computer technology. For us, this is only the beginning. Eventually, we hope to interface with many, if not all, of the local elementary and secondary schools in our area. There are many rural schools throughout the Pacific Northwest. We dream that older kids will be helping, teaching younger kids, regardless of the grade, the time of day, or the location of the school.

Tulane University: Macintosh Computers and Witchcraft

For several years, Professor Richard B. Latner of the Department of History at Tulane University has taught a course on the Salem witchcraft episode. The course is designed to inform students about what happened in Salem, and more important, to teach undergraduate students how to study history. This so-called "inquiry" or "laboratory" approach to teaching history emphasizes developing skills in historical investigation, analysis of information, and written exposition. Students read an abundance of published documentary materials, along with historical works on selected aspects of the Salem events, such as the nature of witchcraft and Puritanism. Throughout the course, students write a series of short essays presenting their responses to problems posed by the instructor.

Recently, a microcomputer component has been added to the Salem course.

Recently, a microcomputer component has been added to the Salem course. Quantitative analysis and the use of computers have been a significant component of historical research and graduate training for a generation, but little of this revolutionary change has filtered down to the undergraduate history curriculum. Much of the Salem documentary material lends itself to quantitative analysis: tax records, committee and church membership lists, petitions for and against the trials, and data about accusers and the accused. For example, names on petitions indicating support or opposition to the trials can be linked with names on tax lists to show the relative economic situation of each group. The Salem material is, therefore, a suitable vehicle for introducing undergraduate humanities students to the techniques and methodologies of quantitative history and the utilization of microcomputers.

A grant from Apple Computer helped to establish a small Macintosh laboratory in the history department. Professor Latner then developed the Salem Data Set using Data Desk, a statistical software package designed for the Macintosh. Data Desk is a powerful, but easy-to-learn program that can be readily adapted to historical analysis. It can quickly create scatterplots, boxplots, and histograms, sort numbers, construct contingency tables, calculate means and medians, and perform other statistical analyses relevant to historical data.

Last spring, Professor Latner offered a trial run of the computer-assisted Salem witchcraft course in a limited-enrollment seminar. In addition to teaching the use of the Macintosh and some basic statistical concepts, Latner constructed a set of exercises that students performed using the Salem Data Set.

A histogram of the numbers of accusations per month, for example, displays the chronological boundaries of the witchcraft episode from February 1692 to October 1692, with only a few accusations occurring between October and January 1693. It also reflects the ebb and flow of witchcraft accusations.

By the end of the semester, all students—even those who had initially resisted working with computers—were familiar with the Macintosh and the Data Desk software. (At least one student took advantage of the computer lab to become adept at word processing, as well.) In spite of the typical problems of a mechanical and substantive nature encountered during this experimental run, the computer-assisted course was successful in introducing technological and methodological innovations to the humanities curriculum.

At least one student took advantage of the computer lab to become adept at word processing, as well.

The Salem Data Set, along with a manual describing its use in the classroom, is now available. The Data Set, though initially designed to supplement a series of traditional assignments dealing with the Salem witchcraft episode, can be used in a wide variety of classroom activities. It can supplement exercises in survey American history or American colonial history courses, or it can be used as a fully integrated microcomputer component for a one- or two-semester course on Salem.

For further information, contact:
Professor Richard B. Latner
Department of History
Newcomb College, Tulane University
New Orleans, LA 70118
(504) 865-5588
BITNET: CBOX @ TCSVM

Astrophysical Research Consortium (ARC): Remote Astronomy and the Macintosh

Background

The Astrophysical Research Consortium (ARC), which consists of five universities (the University of Chicago, New Mexico State University, Princeton University, the University of Washington, and Washington State University), is currently engaged in building a 3.5-meter telescope located at Apache Point on Sacramento Peak in New Mexico. This will be one of the largest privately owned and operated telescopes in the country and will have a number of unique features. Among these, it will be the first large telescope designed from the outset for remote operation. An astronomer may call up the telescope via a telephone or satellite link, check the

weather, open the dome, point the telescope, select one of several on-line instruments, take data, and receive real-time data displays and guiding information. Remote operation of the telescope and instruments will even be required over short distances on-site, as well as by astronomers at their home institutions, since the control room and laboratory facilities will be located in a separate building to minimize thermal effects on seeing within the telescope enclosure.

Through telescience, an astronomer and team can be notified of the conditions and at the terminal within half an hour.

This type of scientific data gathering is becoming known as "telescience," that is, doing science in real time while being physically separated from the actual data-taking equipment. Whether the equipment is sitting in a lab in the next room, or on a satellite in orbit, the methods of doing the science are identical. Telescience is particularly useful in astronomy, not only because of unmanned, astronomical satellites, but also because of the expense of travel to a mountaintop observatory and the gamble of waiting for good viewing conditions. Through telescience, an astronomer and team can be notified of the conditions and at the terminal within half an hour.

The overall goal of remote observing software is to provide enough information, sufficient control, and rapid response capabilities so that observers can function as well as if they were at the site. There are two obvious key ingredients for remote observing: adequate data transmission bandwidth and negligible transmission delays.

In order for the observer to be adequately informed, there must be not only a good interactive environment, but also an ability to easily review both instrumental and housekeeping data. At telescopes, information is available in many forms. In most older computer interfaces, all of these data were usually translated into words or numbers in some sort of tabular display. Control functions were implemented via a "language" that had to be mastered and remembered. However, most telescopes and instruments have been designed to be controlled by switches and control panels, and heavy use is made of analog gauges and annunciator lights. Changes in instrument states (particularly "down" conditions) need to be recognized quickly: this is more easily accomplished by a distinct pattern change in an analog display (or an audible cue) than by a change in the content of a written message.

Graphical interfaces have been used in the past on systems that could accommodate them; this meant expensive minicomputers with expensive graphics terminals and no real built-in graphical tools such as windows, controls, menus, and so on. To generate a graphical interface, there was not only an investment in costly hardware, there was also a considerable investment in software development time. With the introduction of the Macintosh, these major obstacles were overcome. We decided to implement a graphical interface on a Macintosh because of its event-driven environment, built-in toolkit, screen clarity and resolution, and relatively low cost. The event-driven nature of the system meant

that operating modes could be avoided: all functions of the program would be available to the user at any time. We also chose to follow the standard interface guidelines, so that any new user already familiar with the Macintosh would understand most of the options and features.

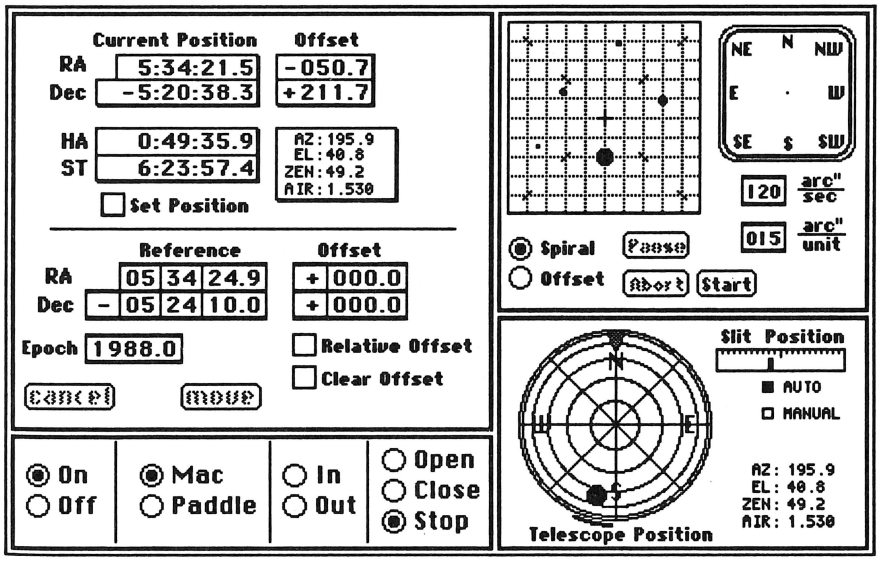
The 41-Inch Prototype

Because the Apache Point Observatory is not yet finished, and to gain experience with graphical instrument control, we have implemented a number of such interfaces to various instruments since 1985. The remainder of this discussion focuses on the current prototype interface to the Yerkes Observatory 41-inch reflecting telescope that is being upgraded for remote use.

In the course of prototyping, we developed several new types of controls. One of these (ThumbWheels) has been especially useful and is now used in most of our applications. The ThumbWheel class allows us to create analogs to hardware thumbwheels: switches that work similarly to the time-setting controls in the Macintosh Control Panel. These thumbwheels can replace the standard Macintosh scroll bar.

In the course of prototyping, we developed several new types of controls.

The prototype interface is shown in the following figure. There are four window "panes" in the current control panel, which are summarized beginning in the upper left and continuing clockwise.



Main display/control—The top half of this pane displays the current coordinates of the telescope and any offset from a reference position selected in the lower half. Also shown are various other parameters dependent on the telescope position. In addition, the sidereal time and hour angle of the telescope are updated every

second. The bottom half of the pane is for data entry. After a reference position is entered, an offset may be added (either relative or absolute) to allow movement about a desired point.

Guider/Tweaker—While most facility observatories have a night assistant to control large moves of the telescope, the astronomer still needs to have control over small motions about some reference point. This control is almost always in the form of a "tweaker box" with four buttons for the four possible directions of motion. The button position on the box provides a tactile clue to its action while the astronomer's eyes are glued to a television output of the guide camera. Because this was impossible with the mouse, we decided to implement nonstandard controls, whose eight outlines are the directional sector areas in the rounded square. However, this graphical tweaker suffers from one of the drawbacks mentioned above: any delays due to satellite links will significantly interfere with the responsiveness of the control.

Perhaps a better type of offset control is shown in the gridded area of this pane. It contains a plot of the objects' positions as sent from the telescope's guide camera computer. The size of each filled circle represents the brightness of the corresponding star. Rather than dialing in a small offset in the main control pane, or using the graphical tweaker, the user may grab any object in the field with the mouse and drag it to the desired offset, sending the message to the telescope to move to that position. This type of control is more intuitive to use than the thumbwheel entry and also gives a graphical map of the telescope movement.

This pane provides two other operations: a spiral search may be initiated if the desired starfield is not immediately found; and autoguiding may be maintained by measuring the position of a designated guide star and automatically tweaking the telescope to counteract any drifts in its position.

Orientation display—An alt-azimuth display provides the user with a polar plot of the telescope's pointing relative to its mount. The concentric circles are projections onto a plane (the observatory floor) for every 15 feet of elevation. The black circle represents the telescope at a particular altitude and azimuth. The slit position of the dome is also shown and is either automatically or manually updated as the telescope tracks the stars.

Power, focus, and opening and closing the slit in the dome are all controllable from the Macintosh.

Observatory controls—The four sets of radio buttons in the last pane simply control various relays in the observatory. Power, computer enable (as opposed to the enabling of a hardware control paddle located in the dome), focus, and opening and closing the slit in the dome are all controllable from the Macintosh.

The Future

While the interface currently runs on both the Macintosh Plus and Macintosh II, we are targeting the actual program for a Macintosh II. This will allow gray-scale (or false color) imaging of any picture from the telescope. At present, picture sizes are about 800 x 800 pixels, requiring a larger screen than the 640 x 480 standard Macintosh II monitor. We have to plan for images growing to 2,000 x 2,000 pixels sometime in the future. Ultimately, we may use two monitors—one for pictures and one for control—or we may substitute one very large monitor for both displays.

We envision a typical observing session with the ARC telescope to be the following: the astronomer calls up the observatory site computer from his or her Macintosh and logs in. A quick check at the weather station instruments provides information about the outside conditions. If everything is fine for an observing session, the astronomer selects an instrument from a menu, the appropriate control panel opens on the desktop, and the corresponding instrument is switched on-line both electronically and optically. The telescope control panel is also opened, which provides all the necessary controls and readouts for operation. A guiding window, which shows a real-time image from a guiding camera on the telescope and allows the astronomer to verify the telescope's position and atmospheric "seeing," is moved to a convenient location on the desktop. By pointing with the mouse, virtually all commands can be sent to the telescope and instrument. Data, either in the form of pictures, or graphs and stripcharts from other detectors, will be displayed in data display windows.

By pointing with the mouse, virtually all commands can be sent to the telescope and instrument.

The interface is still evolving as we continue to evaluate its effectiveness. Many types of controls familiar to astronomers may yet have to be abandoned and new ones invented. In addition to the telescope control panel, any instruments on the telescope must also be controlled by the Macintosh. Integration of various instrument controls into the same Macintosh screen is currently under way. The 41-inch prototype is one of the first steps—we hope to have a working interface for the ARC telescope by early 1989.

For more information, contact:
R. F. Loewenstein
Yerkes Observatory and Dept. of Astronomy and Astrophysics
The University of Chicago
Williams Bay, WI 53191

California State University: Advanced Technology Laboratory

By Francoise Corey and Jerry Sprecher

For the last few years, keeping abreast of changes in technology has been a major challenge for faculty and staff on our nation's campuses. EDUCOM, through activities like the Software Initiative, regional seminars, and conferences, is currently addressing this issue on a national level. In an effort to address similar needs on the 19 campuses of the California State University, the Office of Computing and Communications Resources created the Academic Computing Enhancement (ACE) Institute. ACE facilitates the use of new technology by providing a supportive environment for innovative, technology-related projects. As a result of this effort, the Advanced Technology Laboratory (ATL) was conceived.

The ATL's goal is to provide a stimulating environment that promotes positive experiences with advanced technology.

The ATL's goal is to provide a stimulating environment that promotes positive experiences with advanced technology. In its simplest form, this involves arrangements with vendors and others outside the system to provide new technology very soon after introduction. It also involves dissemination of accomplishments by one faculty member on one campus to the more than 18,500 faculty members in the system.

Specific services provided by the ATL include tours, demonstrations, seminars, and vendor partnership activities. At the broadest level, tours offer participants hands-on experience with a variety of different technologies housed in the lab. Demonstrations serve to narrow the focus and provide more in-depth information about one technology or group of technologies. Seminars and workshops are the most thorough and may be led by ATL staff, CSU faculty or staff, outside educators, or vendor staff. These sessions are presented in the ATL's "classroom of the future" and are designed to provide details about a particular new technology. Often, the sessions result in participants identifying potential applications for their classrooms or offices. Finally, the ATL is used as a support facility to assist faculty and staff who are participating in vendor grant projects or systemwide activities.

The selection of computing equipment to be housed in the ATL is based on the concept of "technology transference," which requires that a vendor partnership be established when it will help CSU reach its goal of introducing potentially useful technology to faculty and staff. The partnership must be mutually beneficial; however, since CSU represents such a large base of prospective users, developers, buyers, testers, identifying "win-win" relationships has not been difficult.

One such relationship is with Apple Computer—a partnership that resulted in our acquisition of 17 fully configured Macintosh II workstations, Macintosh software, AppleTalk networking equipment, and a LaserWriter printer. All workstations will be networked, and, in addition, each Macintosh II will be a component of a video network, allowing the instructor to monitor every student station. Activities taking place at both student and instructor stations can be projected on a large screen.

The Macintosh II was the ideal workstation for the ATL, because it could provide both the standard Macintosh enviroment and the UNIX environment (with the A/UX operating system). In addition, with the planned addition of third-party hardware, it can also provide a DOS environment. Its power, graphics capabilities, and ease of use make it appropriate for a wide variety of educational applications.

The Macintosh II will also be connected to CSU's X.25 wide area network, so that resources on all 19 campuses of the 1,000-mile CSU system can be accessed from the lab—including the supercomputers located in the San Diego Supercomputer Center. Finally, via Telenet, we also plan to make external resources available.

The ATL was designed, in part, as a showcase for state-of-the-art computing technology, but for a very large audience. To reach this large and geographically remote audience, a low-light, high-image video camera will be used to record training workshops, conferences, and vendors' presentations. Recorded sessions of general interest will be disseminated to the 19 CSU campuses. Similarly, sophisticated slide-making equipment is also part of the ATL "dissemination" strategy.

The ATL was designed, in part, as a showcase for state-of-the-art computing technology

As technology evolves, so must the ATL. As a result, new vendor partnerships are emerging. Recently, relationships have been formed with Microsoft, Symantec, and Letraset.

For more information, contact:
Jerry W. Sprecher
Advanced Technology Laboratory
P.O. Box 92289
Long Beach, CA 90809

Tarrant County Junior College: Computerized Freshman Composition

By Eddye S. Gallagher

Imagine 400 students divided into 13 freshman composition sections, each of which is taught in a classroom equipped with 32 Macintosh SE computers and eight ImageWriter II printers. That image has become a reality for six instructors at the Northeast Campus of Tarrant County Junior College.

Project Background

In December 1986, the English department submitted a proposal requesting approval to investigate, evaluate, and plan a computerized classroom. District guidelines required 100 percent cooperation and participation by full-time department members. All 13 department members had to attend ten hours of workshops, which were designed to introduce them to computers, as well as to teach them how to evaluate and select appropriate software and use the computer in a classroom setting. In addition, we had to provide time for establishing criteria and goals for a computerized course.

Our investigation was telling us that IBMs might not be the best machines for our purposes.

During the 1987 spring semester, we conducted workshops with representatives of computer and software companies and our own Computer Services personnel. Our initial plans required us to borrow IBM computers from another department so we could conduct a pilot program. Meanwhile, however, our investigation was telling us that IBMs might not be the best machines for our purposes. Since we teach writing and not computer science, we wanted a course that would allow students to compose all of their writing assignments on the computer. We also wanted students to learn the value of revising work. Therefore, we wanted an easy-to-use word processing package that would be combined with an editing program to help students identify and correct mechanical problems with their writing.

Our investigation led us to the Macintosh computer with MacWrite and MacProof software. We then started a campaign to equip a classroom with computers for the fall 1987 semester. With the support of Computer Services and our division chair, we wrote a second proposal in April, requesting that 32 Macintosh SE computers and eight ImageWriter II printers be installed in a classroom dedicated to freshman composition classes. (Funding would be through bond monies.) By May, we had district approval for fall classes.

We spent our summer vacation selecting furniture, designing floor plans, and working with electricians, painters, and other maintenance personnel. We coordinated placement and installation of equipment and supplies with Computer Services, planned evalu-

ation criteria and documentation with the college's Office of Research, and wrote guidelines, tutorials, lesson plans, and handouts for students.

Classroom Integration

Each semester, we begin with the Macintosh SE "guided tour," then move gradually—through short assignments—into use of MacWrite, and finally, introduce MacProof before the first graded assignment. By that point, most students feel confident with the computer and the software and can concentrate on writing.

We spend a portion of every class discussing the goals and objectives for that day and explaining any points about writing or revising that students need for the current assignment. Then the students spend the remainder of class working on their papers. They seem to appreciate the individual time we can give them as they encounter problems during the writing process. This semester, we added MacView Frame, which helps illustrate lessons. This liquid-crystal display (LCD) works with a Macintosh and an overhead projector to enlarge and project a screen's content.

The students seem to appreciate the individual time we can give them as they encounter problems during the writing process.

Outside of class time, we open the classroom for students to work on their assignments. As part of our office hours, the six faculty members work in this open classroom. The room opens at 7:45 A.M. and closes at 10:00 P.M. (with the exception of one hour) four days a week. To have time for faculty, department, and committee meetings, we close the classroom on Friday afternoons at 12:30. We also have four Macintosh SE computers and a printer in the Computer Learning Center (CLC) in our library, so students have access to those when classes are meeting. (The demand for computers has been so great that the CLC has requested more systems, and some students wait outside the classroom during regular class hours to see if someone is absent so they can ask to use that student's computer.)

Successes

We expected students to love the ease of revising on a computer, and they did. (In fact, one student called me three times one day to substitute versions of her paper because she could not stop revising after she had submitted an assignment. These revisions took place after numerous other revisions in drafting stages before she ever submitted the first copy.) Another advantage was a total surprise. We had been concerned about what would happen to the class's sense of unity when students began to do much of their work independently. Although we planned some peer editing activities to help establish a bonding among students, we didn't need to worry. Students made friends by helping each other with computer or printer problems, and even began peer editing and evaluating practices before we introduced the concept.

We did find one disadvantage. Students no longer wanted to listen to "boring lectures"; they wanted to spend class time writing. Although few class sessions were totally lecture oriented, students wanted *all* the class time for composing and revising.

Evaluation

During spring registration, students specifically asked for computerized composition.

As part of our evaluation plan, during the first and last weeks of the semester, we administered the College Board's Written Expression Test to all our freshman composition students. We also prepared a questionnaire that asked students about their writing habits and attitudes. Although the research isn't complete on the attitude survey, computerized composition students raised their scores by 1.4 points of 40 (28.0 to 29.4) as compared to 0.9 and 1.3 (26.8 to 27.6) improvements seen by students in the other two freshman composition options (traditional classroom settings with no computers). We also learned that the class was successful in a more intuitive way, because in the fall, our computerized sections filled rapidly. And during spring registration, students specifically asked for computerized composition. In addition, the students who worked with computers in the fall requested a second semester of computerized composition. (Second-semester composition students have conventional classes.)

Fortunately, we have experienced no major problems in maintenance and funding. In fact, we have seldom had a computer or a printer out of commission for more than a few hours. For the most part, problems came from students trying to copy the software onto their own disks or rearranging the contents of folders on the hard disk. That situation was remedied by the addition of software that prevents anyone from copying or moving items from the hard disk

We consider ourselved fortunate to have a strong support system in our Computer Services area. They provide maintenance support for equipment and help with software problems. In addition, they furnish our paper, ribbons, and software updates. Help is only a phone call away, and problems are generally solved within an hour.

We have been so pleased with the computer classroom concept that we have requested that another classroom be computer-equipped to serve developmental and technical writing students.

For more information, contact:
Eddye Gallagher
English Department
Tarrant County Junior College Northeast Campus
828 Harwood Road
Hurst, TX 76054

Texas Christian University: Vertebrate Paleontology, Geology, Geostatistics—and General Hacking

Introduction

As an evangelist-without-portfolio, I have been exposing geologists and other scientists to the wonders of the Macintosh user interface since 1984. Access to the Macintosh Toolbox through a variety of programming languages (I prefer to use BASIC) and a wealth of versatile programs has allowed me to produce custom software and templates, both for my own research and for use by students. Some of the applications I have developed are for modeling the mechanics of jaw muscle systems, for interactive calculation of the mechanical properties of beam cross sections, and for digitizing maps to convert them into PICT files. I have also developed a variety of Trapeze templates for statistical and graphical analysis. Currently, my two most important development tools are BASIC and Trapeze.

BASIC

I have primarily used the Microsoft BASIC interpreter and compiler for developing applications. MS-BASIC 3.0 supports procedures with local variables, a variety of QuickDraw and toolbox calls at a high level, and external libraries. The compiler also supports a CASE structure. The largest problem with MS-BASIC is the lack of extensive toolbox support without external libraries, and the lack of long integer variables and long or word-length PEEK and POKE. As my programs became more complicated, I badly needed extensive full-speed toolbox support, and I began to have difficulties. Since further development of the MS-BASIC compiler is in limbo, I have started to work with Z-BASIC.

Z-BASIC, by Zedcore, is an integrated BASIC development system that, like the MS-BASIC compiler, produces stand-alone applications. Its most glaring defects are the absence of procedures with local variables and a somewhat limited editor. These are, however, far overshadowed by its extensive toolbox interface, the fast and compact object code it produces, its excellent debugging facilities, and its support of long integer variables and extended PEEK and POKE functions. It also supports long functions, CASE statements, DO-Until loops, and scroll bars. Z-BASIC provides full or partial interfaces to MacinTalk, QuickDraw (including indirect color QuickDraw support on a Macintosh II), and the SCSI, AppleTalk, Resource, File, Window, Control, Menu, TextEdit, Dialog, Scrap, Memory, System Error, and List Managers and Toolbox utilities and Segment loader.

Z-BASIC is an integrated BASIC development system that produces stand-alone applications.

Trapeze

I find that one of the major tools I now use for custom applications is Trapeze, developed by Data Tailor. Originally marketed as a spreadsheet, this free-form data analysis worksheet system has become an integral part of my library. I have written Trapeze worksheets that perform linear discriminant analysis, first- through third-order trend surface analysis with gray-level maps, file translation, reduced major axis line fits, and more. Trapeze uses a block-oriented approach that is fundamentally different from that of standard spreadsheets. It includes support for matrix algebra, Gaussian solution of linear algebraic systems, two types of integration, and even a single function that returns an entire amortization sheet. It offers "standard" graphics, as well as gray-level maps (which can be stored in PostScript format for rapid printing on a LaserWriter) and polar diagrams. The scatterchart function is flexible enough to allow plotting of three-dimensional grid surfaces and geologically useful triangular diagrams. Unlike the separate graphics sheets of Microsoft Excel, Trapeze graphics blocks can be placed next to data blocks on the same worksheet in any position and can be enlarged to any size.

Jaw Muscle Simulation

One of my areas of interest is vertebrate biomechanics with an evolutionary focus. I like to use biomechanical models to evaluate the evolution of mechanical structures in vertebrates, particularly reptilian feeding systems. Thirteen years ago, I wrote the first version of a program to do simple two-dimensional modeling of the mechanics of jaw systems. The user had to make some measurements from a skull involving the position of muscle attachments and general muscle lengths with the jaws closed. The model then numerically opened and closed the jaws, printing out a long table of information, such as the instantaneous length and maximum moment arm of the muscles and relative vertical and horizontal loading components at the jaw joint.

What the program lacked, until the advent of the Macintosh, was interactive capability for both the entry and display of results.

What the program lacked, until the advent of the Macintosh, was interactive capability for both the entry and display of results. I had the first version ported over to my Macintosh 128K in early 1985 in MS-BASIC 1.0, and began to rewrite the program to take advantage of menus, buttons, and the mouse. When Clear Lake announced its ToolLib with enhanced toolbox support, I purchased it and finished the conversion. When the compiler beta-test began, I started to get the program running as a stand-alone application, which significantly improved program performance and allowed me to give the application and its data files unique icons.

This program automates a previously time-consuming and laborious task and allows interactive modeling of the mechanics of jaw systems. It certainly demonstrates that the Macintosh interface can be used to reduce the tedious part of entering data and increase both the throughput and fun of modeling.

Trend Surface Analysis

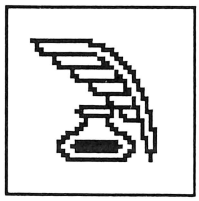
In geology, it is frequently useful to find the mathematical surface that best fits a series of data points taken on some real surface. One of the simplest techniques is trend surface analysis, in which techniques similar to linear regressions are used to find the surfaces. The basic mathematics involved are fairly simple and can be calculated by a sequence of matrix operations on the data. Many canned programs exist for calculating trend surfaces, and a few now exist for the Macintosh that will calculate trend surfaces and contour them. I wanted to make use of the graphics capability of the Macintosh to generate gray-level (or density-sliced) maps of the surfaces where the shade of gray was a function of the altitude of the surface. I also wanted an extremely interactive program that would display both the statistical results of the fit and the map.

Trapeze, with its gray-level/color-mapping CONTOUR function and built-in matrix manipulation functions, was a natural for this kind of problem. Trapeze can also use the 68881 floating-point coprocessor, which further increases throughput. Two versions were written, one with four separate sheets to avoid out-of-memory problems on the students' 512K Enhanced systems, and one with the data input and analysis on four pages of a single worksheet for more powerful Macintosh computers. The description below is based on the student version. One of the sheets is the data entry sheet. Here data can be entered from the keyboard, pasted in from the Clipboard, or imported from a text file. Three analysis sheets are linked to the main input sheet: one each for the first-, second-, and third-order trend surfaces. For a particular analysis, one needs merely to load the analysis sheet and then instruct Trapeze to load any linked sheets. Each analysis sheet has the necessary matrix command blocks (program) to extract the original data, perform a pair-wise deletion of missing data and the required calculations, display an analysis of variance table for statistical analysis of the surface fit, and a gray-map of the resulting surface.

Further analysis can be performed by writing new Trapeze templates to link to the original sheet. In addition, students can easily go through the blocks used to compose the worksheets to see how matrix algebra can be used to solve trend surface problems. No other product currently available on the Macintosh could allow such an analysis with mapping.

For more information, contact:
Dr. Arthur B. Busbey III
Department of Geology
Texas Christian University
Ft. Worth, TX 76129
(817) 921-7270

News and Notes



The **News and Notes** section contains information of general interest to those in the Macintosh community. If you have information or know about events that you feel fit into this category, please send your news along.

Joint Development Effort with DEC

Digital Equipment Corporation and Apple Computer, Inc. announced a joint development effort to integrate Macintosh personal computers and AppleTalk networks with VAX systems and DECnet/OSI networks. These development efforts will take advantage of open standards for desktop integration, based on the industry-standard Open Systems Interconnection (OSI) model of the International Standards Organization (ISO).

The collaboration between the two companies represents one of the strongest alliances yet in the computer industry, bringing together the best technologies from two of the industry's best-known innovators. The big winners will be large companies and universities, which will have new opportunities for powerful, highly integrated, multivendor systems that are also easy to use.

The big winners will be large companies and universities, which will have new opportunities for powerful, highly integrated, multivendor systems that are also easy to use.

"Digital's strategy is to provide networked server and desktop solutions," said Kenneth Olsen, president of Digital Equipment Corporation. "These include the VT terminal family, and VAXmates and VAXstation systems, designed to allow users with different needs to share resources, information, and applications across a network based on international standards. Digital also integrates other quality desktop systems—the development efforts with Apple will extend that integration to our customers who use and love the Macintosh, and will advance the movement toward the OSI international standards."

Apple's Perspective on the DEC Agreement

In the following interview, Apple C.O.O. Del Yocam discusses some specifics about the alliance:

Q: What is the overall goal of the alliance with DEC?

DY: Both Apple and Digital want to make it possible for our respective products to work together more effectively in the same environment. Joint development efforts will make it simpler for current and future customers—as well as value-added resellers (VARs), Integrated System Vendors (ISVs), and third-party developers—to use Apple and DEC products together.

Q: Why is this alliance important for Apple?

DY: Most important, the alliance is good news for our customers. The joint development products will make it easier for Macintosh users to access the VAX environment, and it will give customers in the DEC world unparalleled desktop control over information that resides on a VAX. The alliance also enhances our standing in the business, government, and higher education marketplaces. Digital is an acknowledged industry leader in supplying solutions to these groups. The credibility and seriousness of our commitment to these customers is therefore strengthened by our joint development association with Digital. So you could say it represents a vote of confidence for our networking and communications strategies.

Q: Is the alliance evidence that either company plans to take an equity position in the other? Is there talk of a merger?

DY: None whatsoever. We definitely plan to remain independent...and so does Digital.

Q: Is this announcement really an effort to compete cooperatively against IBM?

We're engaging in joint development efforts because our customers tell us that's what they want.

DY: No. We make decisions based on what makes sense for our customers. We're engaging in joint development efforts because our customers tell us that's what they want.

Q: Who's going to be developing and implementing the products or capabilities that come out of the alliance? Does the agreement indicate that Apple is going to share resources, staffs, independent business units, or things of that nature?

DY: All the development will be accomplished through cooperative efforts within the existing product development structures of Digital and Apple. That means we'll specifiy designs together, but each company will implement products and capabilities independently. So Apple will develop products and capabilities for AppleTalk; Digital for DECnet.

Q: When are we going to start seeing the results of the joint development? Can you give us an idea of what they will be?

DY: The initial implementation will be a VAX-based DECnet gateway for AppleTalk.

Apple Acquires Network Innovations

Apple has acquired Network Innovations Corporation of Cupertino, California. Founded in 1984 by Jim Groff and Paul Weinberg, Network Innovations develops and markets standard connectivity software products for distributed computing environments. This acquisition reinforces Apple's ongoing commitment to networking and communications.

With the NI team (the company presently has eight employees), Apple will be developing and promoting CL/1, Network Innovations' connectivity language, as a new industry standard across all major computing environments. CL/1 is a universal, high-level, SQL-based data access language for software developers that provides transparent access from desktop applications to data on host systems. The CL/1 language is designed to operate over the industry-standard protocols supported by Apple products, including AppleTalk, DECnet, OSI, SNA 3270, and APPC/LU6.2.

Network Innovations has initially focused its attention on VAX/VMS systems and, in the future, will develop similar capabilities for connecting Macintosh to IBM mainframe databases. Additionally, Network Innovations will develop and market CL/1 toolkits to support DOS and OS/2 workstations. When this strategy is fully implemented, information systems managers will have a very elegant way to connect Macintosh to both Digital and IBM host data, while taking full advantage of the Macintosh user interface.

Network Innovations will continue to operate as an independent company, with Apple representatives on its board of directors. The companies will cooperate closely to develop and market new connectivity solutions.

University of Illinois Wins Apple's Project 2000 Contest

The University of Illinois in Urbana-Champaign won first-place honors in Apple Computer's Project 2000 contest—a competition designed to encourage students to explore the future of technology and share their vision of personal computing in the year 2000.

The winning design, "Tablet: Personal Computer of the Year 2000," integrates the power of computation and communication into a package as compact and portable as a notebook. Members of the winning team were Bartlett Mel, Arch D. Robison, Steven Skiena, Kurt Thearling, and Luke Young. Faculty advisors were Stephen M. Omohundro and Stephen Wolfram.

The winning design integrates the power of computation and communication into a package as compact and portable as a notebook.

Second place was awarded to Princeton University for its design "The Apple PIE" (Personal Information Environment), which integrates the portable radio, television, telephone, and other information technologies that are already widely accepted. Third place was awarded to the University of Minnesota for its design, "CORE," which lets users carry the core of their computers—memory, processing power, learning capability, and the accumulated knowledge—wherever they go.

Each design was judged on its creativity, its use and support of technology, and its value to society.

Finalists presented their papers to a panel of distinguished judges, including futurist Alvin Toffler, science fiction writer Ray Bradbury, education author Diane Ravitch, and computer pioneers Steve Wozniak and Alan Kay. Each design was judged on its creativity, its use and support of technology, and its value to society.

The contest began last September, when Apple extended an invitation to 12 leading public and private universities (representing a geographic and demographic cross section of institutions making extensive use of computers. Schools had the option of incorporating the contest into their curriculum or offering it as an extracurricular activity.

Student teams consisting of up to five members and a faculty advisor were asked to describe the design of the personal computer of the year 2000 in a paper no longer than 20 pages. An unlimited number of teams were eligible to participate; however, only two entries could be submitted to Apple from each school. The papers were to include the purpose of the product, the technologies team members believed would be available in the year 2000, and the way in which they would be used.

The five finalist teams and their faculty advisors received an all-expenses-paid trip to Apple's corporate headquarters and a tour of the company's engineering departments.

Each student on the first-place team received $2,000 to be applied toward the purchase of Apple Computer products, as well as an invitation to accept a paid summer internship at Apple. Graduating members of the team will be given the opportunity to interview at Apple for full-time employment. The winning school will receive a Macintosh laboratory, consisting of five Macintosh SE computers and one LaserWriter printer, connected by the AppleTalk Network System.

Second-place team members received $1,000 each, and third-place winners $500 each, to be applied toward the purchase of Apple Computer products. Faculty advisors from each participating school were awarded a desktop publishing system, consisting of a Macintosh SE, LaserWriter printer, and appropriate software.

Desktop Publishing Conference

On May 5, 1988, there will be a teleconference concerning desktop publishing. The intended audience is instructors of writing, journalism, graphic arts and design, communication media, and computer science, education administrators interested in curriculum planning, and others interested in desktop publishing.

The teleconference is designed to consider the impact of desktop publishing on the production of written communications—and the broader implications of this impact. The organizers believe that it's time for educators to take a serious look at this innovation: how it evolved; exactly what it is and what it can do; why it is so successful; and what they can do to give students the diversity of skills that desktop publishing requires. This program is funded in part by the U.S. Department of Education's Fund for Improvement of Post-Secondary Education (FIPSE).

The teleconference is designed to consider the impact of desktop publishing—and the broader implications of this impact.

"Desktop Publishing—Changing the Workplace" will be presented as a live (one-way video, two-way audio) satellite teleconference. Viewers will attend the teleconference via satellite-receive facilities, and can call in via telephone lines that will be provided especially for that purpose. The live panel discussion-interaction component of the presentation will be juxtaposed with pretaped video segments. These segments will "transport" the viewers to situations where the innovative use of desktop publishing is in full swing, to see a variety of examples of desktop publishing in use and meet the people (users and developers) behind this new technology.

The teleconference moderator is Rob Phillips, porofessor, Department of Journalism, Oregon State University. Dr. Phillips is a specialist in mass and interpersonal communications, and has worked as a discussion leader for a variety of televised presentations.

For details about the teleconference, contact:
NUTN Central Office
332 Student Union
Oklahoma State University
Stillwater, OK 74078-0653
(405) 624-5191

To get in touch with the conference producers, contact:
Jon Root, Executive Producer; or
Rick Clingman, Producer/Director
Oregon State University
Kidder Hall 109
Corvallis, OR 97331
(503) 754-2121

Resources in Special Education and Rehabilitation

Decision makers in special education and rehabilitation are asking how microcomputers can enhance the quality of life for disabled children and adults.

This collection of information contains descriptions of more than 1,200 hardware and software products, organizations, and publications that support disabled computer users.

Apple Computer Resources in Special Education and Rehabilitation represents one of the most comprehensive responses currently available. Produced by Apple Computer's Office of Special Education, this collection of information contains descriptions of more than 1,200 hardware and software products, organizations, and publications that support disabled computer users. It is an indispensable resource for anyone responsible for providing computer-based solutions for disabled individuals. The book's catalog number is G6500Z, and it is available for $19.95.

To order, contact:
DLM, Inc.
P.O. Box 4000
Allen, TX 75002
800-442-4711 (Texas)
800-527-4747 (elsewhere in the United States)

EDUCOM Brochure on Software Use

The EDUCOM/ADAPSO brochure, *Using Software,* is the first in a series of educational materials designed by the EDUCOM Software Initiative to help institutions respond positively and realistically to concerns about the ethics and legality of copying software. The brochure was developed cooperatively by representatives of higher education and software vendors, at meetings sponsored by the EDUCOM Software Initiative. It was written to be read by academic users of software—most of whom have little or no understanding of copyright law.

The brochure is visually striking—easy to spot and easy to read. Its purpose is not to elucidate the fine points of the law. Instead, the text cuts through ambiguities to present ethical, legal, and economic issues relevant to users in straightforward language that is informative and educational rather than punitive in tone. It respects and addresses the unique ethical perspective of the educational community. Its central message takes the form of a statement of principle for academic software use that's modeled on an academic honor code:

Respect for intellectual labor and creativity is vital to academic discourse and enterprise. This principle applies to works of all authors and publishers in all media. It encompasses respect for the right to acknowledgment, right to privacy, and right to determine the form, manner, and terms of publication and distribution.

Because electronic information is volatile and easily reproduced, respect for the work and personal expression of others is especially critical in computer environments. Violations of authorial integrity, including plagiarism, invasion of privacy, unauthorized access, and trade secret and copyright violations, may be grounds for sanctions against members of the academic community.

The brochure also provides concise, simple answers to legal questions raised by situations students may regularly face: a friend's request to borrow a spreadsheet, or the desire to copy software available on public campus facilities, for instance. The answers consciously avoid debate over legal ambiguities; instead they provide simple, practical advice that enables students to get on with their work. Finally the brochure recommends several economically feasible alternatives to illegal copying, including shareware, freeware, and institutional discounts.

The brochure provides concise, simple answers to legal questions raised by situations students may regularly face.

In the summer 1987 issue of *Wheels for the Mind,* we printed an order form for this brochure. If you have already ordered *Using Software,* the EDUCOM people would like to know about your dissemination of the brochure and the impact that it has had on your campus.

Please send your information to:
Pauline Eden
EDUCOM Software Initiative
P.O. Box 364
Princeton, NJ 08540
BITNET: ESI@EDUCOM

1988 ACM Scholastic Programming Contest

The Association for Computing Machinery sponsors an annual ACM Scholastic Programming Contest that brings together winners of 12 regional contests to compete for awards, prizes, and scholarships. This year, the event was held in conjunction with the annual Computer Science Conference of the Association for Computing Machinery at the Westin Peachtree Plaza Hotel in Atlanta, Georgia, on February 24.

Twenty-four teams of students and a team of guest celebrities (faculty members, sponsors, and former contestants), representing universities from the United States, Canada, and Europe, competed to try to solve eight programming problems in six hours. This year's results were as follows:

1. California Institute of Technology
2. Carnegie-Mellon University
3. University of Pennsylvania
4. Harvard University
5. Johns Hopkins University
6. University of California at Los Angeles
7. Washington University
8. Purdue University
9. California State University, Sacramento
10. University of Texas at Arlington
11. Drexel University
12. Stanford University
13. State University of Ghent
14. Free University of Amsterdam
15. Rensselaer Polytechnic Institute
16. Texas Tech University
17. University of Maryland
18. University of South Carolina
19. University of Alberta
20. University of Wisconsin at LaCrosse
21. Northern Arizona University
22. Manhattan College
23. University of Florida
24. Drew University

The contest was hotly contested, with the first team solving all eight problems.

The contest was hotly contested, with the first team solving all eight problems, the second and third teams solving seven problems, and the next three teams solving six problems.

The contestants used Macintosh Plus systems with 20-megabyte hard disks running Lightspeed Pascal (provided by Think Technologies, Inc.). Scholarships were awarded by the Upsilon Pi Epsilon Honor Society for the computer sciences. Other contest sponsors included SAMNA Corporation, the ACM Student Chapters, SIGDA, and SIGPLAN.

For more information, contact:
Win Strickland
Georgia Institute of Technology
School of Information and Computer Science
Atlanta, GA 30332
(404) 894-3086

Philadelphia Is Nuts About Hypermedia

June 1–3, Philadelphia, the city of brotherly love, will be the site of the MacAdemia℠ conference, for which—in an appropriately brotherly fashion—Drexel University and the University of Pennsylvania will act as cohosts.

This year's theme is hypermedia, and some 500 educators from the East Coast are expected to attend the conference presentations on innovative and state-of-the-art hypermedia development activities. Drexel will open MacAdemia on June 1 with a brunch and a general session in the main auditorium of its historic main administration building. Throughout the event, faculty members from many schools will demonstrate applications they have developed using the Macintosh and a variety of media. Drexel's demonstrations will include the materials it has used for its Adult Literacy Program, conducted in conjunction with the Mayor's Commission on Literacy in Philadelphia.

This year's theme is hypermedia, and some 500 educators are expected to attend.

A tentative list of hypermedia topics to be presented at MacAdemia includes administration/operations, desktop publishing/graphics, engineering/design, basic sciences, business, math/computer science, networks/connectivity, language and composition, and fine arts/humanities/social sciences. Both universities are also hosting tours of their computing facilities. The universities are easily accessible by plane, train, or car. (Amtrak's 30th Street Station in Philadelphia is only two blocks away from Drexel.)

For more information, contact:
Dr. Jan Biros
OCS Coordinator of the Microcomputing Program
(215) 895-2667

Attention Language Teachers

The Middlebury College Language Schools, in cooperation with CALICO (The Computer Assisted Language Learning and Instruction Consortium) and PICS (The Project in International Communication Studies), will host the 1988 CALICO Summer Institute, a five-week program of workshops and conferences for language teachers on the uses of computers (primarily Macintosh) and video in the teaching of Chinese, French, German, Japanese, Russian, and Spanish. The program will combine introductory, hands-on experience with computers and video, with two specialized conferences for more advanced workers.

Tentative dates are as follows:

Introductory Computer Workshop	July 7–20
Advanced Computer Conference	July 21–26
PICS Advanced Video Conference	July 27–31
Introductory Video Workshop	Aug. 1–13

For more information or application materials, contact:
The New Techniques
Middlebury College Language Schools
Middlebury, VT 05753
(802) 388-3711, ext. 5536

Adding Memory to Your Macintosh

Many Macintosh users are considering adding more memory to their machines. If you are interested in expanding the RAM of your computer, then the following may be of help to you in your planning.

The Macintosh Plus has the following possible configurations:

- 512K, using two 256K SIMMs
- 1 megabyte, using four 256K SIMMS
- 2 megabytes, using two 1-megabyte SIMMs
- 2.5 megabytes, using two 1-megabyte SIMMs, and two 256K SIMMs
- 4 megabytes, using four 1-megabyte SIMMs

It is important to place the SIMMs in the correct location when using a combination of SIMM sizes.

It is important to place the SIMMs in the correct location when using a combination of SIMM sizes, as in the 2.5 megabyte example, and to make sure the right resistors are cut.

The Macintosh SE configurations are the same as those of the Macintosh Plus, but the physical locations are different.

Adding Memory to the Macintosh II

You must always upgrade the Macintosh II in four-SIMM chunks, because of its 32-bit data bus. The eight SIMM connectors are divided into two banks of four, called Bank A and Bank B. Bank A is the bank closest to the edge of the board. In the Macintosh II, unlike the Macintosh Plus and Macintosh SE, there are no resistors to cut; you need only install the SIMMs in the correct place, and you'll be up and running. The possible configurations are as follows:

- 1 megabyte, using four 256K SIMMs in Bank A
- 2 megabytes, using eight 256K SIMMs in Banks A and B

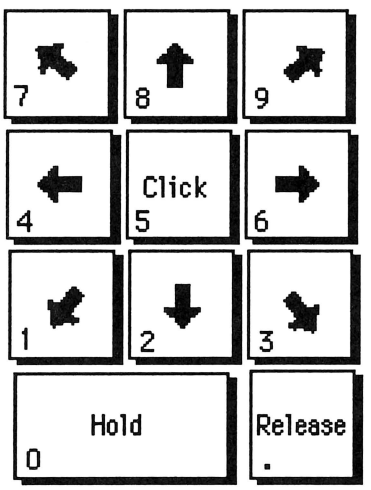
- 4 megabytes, using four 1-megabyte SIMMs in Bank A
- 5 megabytes, using four 1-megabyte SIMMs in Bank A, and four 256K SIMMs in Bank B
- 8 megabytes, using eight 1-megabyte SIMMs in Banks A and B

Again, it is important to make sure the appropriate SIMMs are in the correct bank; when you are using a combination of SIMMs, the larger SIMMs must be in Bank A. This is also true when you are using only four SIMMs.

Easy Access to the Macintosh

Easy Access is a feature that can be found in the latest Macintosh System Folder. It was designed to help disabled people with their use of the Macintosh. One feature turns the keypad into a functional mouse, and another feature provides a way to input complex key sequences more easily.

Easy Access was designed to help disabled people with their use of the Macintosh.



The feature that makes the keypad function as a mouse is called Mouse Keys. This feature is "activated" by pressing Command-Shift-Clear simultaneously. You can then use the numbered keys to move the cursor in eight possible directions. The 5 key on the keypad is equivalent to clicking the mouse button; the 0 key is equivalent to holding the mouse button (as with dragging); and the decimal point is equivalent to releasing the button. To make this feature work, you will need to have a Macintosh Plus, Macintosh SE, or Macintosh II keyboard.

The second feature allows key combinations to be typed sequentially rather than simultaneously. This feature is activated by pressing the Shift key five times in a row.

Help Sought for Decision Support System

I am currently involved in developing a prototype "decision support system" in a Macintosh environment that, if successful, will presage a full implementation within this state. As a result, I am most interested in locating others in the "Macintosh world" who are involved in the concept of *shared decision support systems,* as well as *their effect on group dynamics in a shared network.*

This information and/or contacts could take many forms. If you are interested in sharing such information, or participating in any way, please contact:
Andrew B. Lefton, M.A., Project Director/Case Manager
Institute for Child Health Policy
State University System Institute at the University of Florida
5700 S.W. 34th Street, Suite 323
Gainesville, FL 32608
(904) 392-5904

ICEC/Apple Fellowship Program

The purpose of the program was to foster the development of software important to the goals of both ICEC and Apple.

In June of 1987, Apple Computer, Inc. announced a major grant program for member schools of the Interuniversity Consortium for Educational Computing (ICEC). The purpose of the program was to foster the development of software important to the goals of both ICEC and Apple.

Each participating school was awarded a Macintosh II system and a student fellowship that included a $2,500 stipend. The fellowships were awarded on a competitive basis by each school in mid-December of last year.

Targeted areas of research included user interface, distributed computing, graphics and animation, speech/sound, multimedia learning, electronic document architecture, operating systems, standards, and networking and communications.

All projects will be completed by the end of July. In August, Apple will host a conference of all fellowship recipients for demonstration purposes.

We at *Wheels for the Mind* congratulate the recipients of the first ICEC/Apple Fellowship Program grants. Listed below are the 1987 awardees and project descriptions:

Recipient
Claudette FongKong-Mungal, Boston College
Project Description
HyperCard: Hypermedia Database:
To use HyperCard to develop a prototype of a hypermedia database intended for two kinds of people: those who will use it as a structural pattern for designing courseware, and those who want to explore the hypermedia enviornment. The specific focus will be on American history, literature, and geography and the connections across those areas.

Recipient
Loretta Reiss, Division of Engineering, Brown University
Project Description
Deaf Speech Instructional Programs:
A diagnostic program to evaluate the speech of deaf people through the use of MacAdios. Features displayed include fundamental frequency, intensity, spectrograms, spectra, nasal emission level, formant frequencies, and indications of timing of rhythm of speech.

A diagnostic program to evaluate the speech of deaf people.

Recipients
Jonathan D. Brown and Larry Lichten, Ph.D., Computer Science Department, California State University—Northridge
Project Description
Machine Interface for Natural Language Development (MIND):
To demonstrate the value of Conceptual Dependency (CD) theory in a real-world environment. Software and interfaces necessary to drive an IBM 7565 robotic manufacturing system using natural language as a two-way man-machine interface are being created.

Recipient
Stuart Shapiro, Department of History, Carnegie-Mellon University
Project Description
Nuclear Plant Simulation:
To produce a simulation of a commercial nuclear power plant to demonstrate to students the difficulty of making decisions in complex environments.

Recipient
Alexander Smith, Ph.D., Music Department, City University of New York
Project Description
Object Music:
A flexible, multilayered interface for direct synthesis of sound, using a Macintosh computer for output, or as a link to professional-quality digital-to-analog conversion devices. A set of optimized primitives to perform the actual synthesis allows users to test their work interactively.

Recipient
David L. Korn, Department of Science and Natural Resources, Cornell University
Project Description
Graphic Visualization Tools: A New Way to Look at Resource Planning:
To develop a library of graphic software learning tools for resource planning problem solving that will draw ideas from existing VAX software.

Recipient
Earle R. Horton, Thayer School of Engineering, Dartmouth College
Project Description
Graphical Simulation Program:
To use the Macintosh graphics interface to create an animated theoretical model of a pulsar, which will be coordinated with a simulation of the radiation pattern observed at radio telescopes on Earth.

To use the Macintosh graphics interface to create an animated theoretical model of a pulsar.

Recipient
C. M. Huang, Department of Mechanical Engineering and Mechanics, Drexel University
Project Description
Geometry Modeling and Post-Processing Routines for Finite Element Analysis Programs on the Macintosh II Computer:
To develop graphics routines for modeling the geometry based on the input data of finite-element mesh, and post-processing routines for generating contour plots for FE analysis programs on an Apple II.

Recipient
Jeffrey S. Chavis, Howard University
Project Description
Howard University Process and Device Design Program (HU-PRODD):
To develop a user-friendly Macintosh software package for process design and analysis of integrated circuits. The program allows for the final semiconductor processing sequence to be developed through computer simulation.

Recipients
Gloria Caballero and Sandra Belkin, Iona College
Project Description
ECOUTEZ ET REGARDEZ:
To develop an interactive, animated, colorful French picture dictionary with built-in voice capability that introduces students to the proper pronunciation of individual words and sentences.

Recipient
Steve Klaass, Maricopa Community College
Project Description
Student Designed Study Aids (SDSA):
To use HyperCard to create a new application for producing the SDSA, enabling students to become involved in surveying the major areas of a discipline, as well as major topics within those areas. The application will allow students to construct summaries and outlines, create flashcards, design definitions of terms, and apply their knowledge to solving concrete problems.

Recipient
Stephen Ervin, School of Architecture and Planning, MIT
Project Description
A Constraint-Based Graphical Editor for Environmental Design:
Constraints are another word for relations—relations between elements that control the behavior of those elements. The application of constraints to a graphics editor is a logical step, especially for some of the more abstract constraints that are more difficult to implement, such as surrounds, touches, and radiates.

Recipient
Robert Marsanyi, Center for Contempory Music, Mills College
Project Description
Hierarchical Music Specification Language (HMSL):
To implement a new set of classes in HMSL to utilize the Macintosh II computer's hardware sound-producing capabilities and build a set of classes that implement a library of sound generation and manipulation algorithms for use in software synthesis. The HMSL is an existing object-oriented computer language developed for use in contemporary music as a composition, performance, and research tool.

Recipient
William J. Sproule, Jr., Pennsylvania State University
Project Description
A Computer Based Landform Design Tool:
A software tool to allow students to better comprehend the technical engineering aspects of grading design, as well as the more artistic sculptural aspects, through interactive three-dimensional simulation of natural and man-made landforms.

A software tool to allow students to better comprehend the technical engineering aspects of grading design

Recipient
Thomas Hendron, Department of Mechanical and Aerospace Engineering, Princeton University
Project Description
Computational Fluid Dynamics Program:
To use a finite difference scheme to solve the equations of motion for a fluid. The data from this program can be downloaded and manipulated on the Macintosh II and will allow the internal combustion engine user to see the computational grid, the values of one of the 12 flow variables at each node by one of three means (color shading, drawing of constant value contours or by vector arrows), and to animate an engine cycle while displaying the values of a particular flow variable at each time step.

Recipient
David Fedchenko, Reed College
Project Description
Rhythmic Notation:
To define a method of musical notation that will not be biased toward any particular rhythmic patterns. The application will provide an environment for the entry, playback (through MIDI), and printing of rhythmically complex music that cannot be reasonably represented using traditional notation.

Recipient
Chris A. Borkowski, Mechanical Engineering Department, Rensselaer Polytechnic Institute
Project Description
Use of a Computer in a Senior Level Experimentation Course:
To apply HyperCard to a tutorial presentation, simulation, and operation of an advanced engineering classroom experiment. The emphasis of this project will be to develop complete courseware, based on HyperCard, for two experiments: Heat Exchanger Performance Analysis and Forced Convection over a Flat Plate.

A CAI program on biological aspects of a local marsh.

Recipient
Jeff Hilton , Biology Department, Southwestern College
Project Description
Animal Inhabitants of the Sweetwater Marsh:
To develop a CAI program on biological aspects of a local marsh, including historical, ecological, and research implications of this area. The program consists of relational stacks that include the common name, scientific name, and photographic image, as well as general information about each subject.

Recipients
Richard Voyles and David M. Cannon, Mechanical Engineering Department, Stanford University
Project Description
To develop graphics-based methods for entering, altering, and displaying finite-element structures data on a Macintosh II. Voyles is concentrating on developing the algorithms and programming the graphics display and user interface for the Macintosh II. Cannon is focusing on direct application of the graphics/user interfaces to specific structural analysis problems.

Recipient
Brian Reilly, School of Education, Division of Language and Literacy, University of California—Berkeley
Project Description
Hypermedia Environment to Deliver Reading and Study Skills:
To construct a hypermedia environment to deliver reading and study skills instruction to students in their freshman year. The hypermedia implementation will offer access to all of the text-based materials currently in use at the Learning Skills Center and to a series of videotapes of classroom instruction.

Recipient
Dan Roth, University of Delaware
Project Description
A Prolog Engine for Latin Form Analysis:
To complete a Prolog engine for the recognition and generation of inflected Latin words. Once implemented on the Macintosh II, the engine will become the core of a data entry utility.

Recipient
Carla Scaletti, Music Department, University of Illinois
Project Description
Kyma: An Object-Oriented Language for Music Composition:
To develop various graphic displays of synthesized and/or processed-live signals computed in real time on the Platypus signal processor and displayed on the Macintosh II. An additional focus will be to develop multiple representations for each abstract sound structure, in which changes to the graphic representation result in changes to the sound structure being represented.

Recipient
Gregory C. Braunberg, Department of Mechanical Engineering, University of Maryland
Project Description
Infrared Thermography on the Macintosh II
Infrared thermography is a technique for measuring the temperature distribution across an object using a heat-sensitive camera. This thermal "picture" can be stored as a computer image, analyzed and manipulated. Experimentally obtained thermographs can be used to check the validity of theoretical heat-transfer models.

Recipient
Mathew Barritt, University of Michigan
Project Description
A Color Reification Aid for Decision-Making (CRAM):
A color display is used to help people intuitively "grasp" the balance of meaning represented by a set of variables in a complex decision (a concept developed by Fred Goodman).

A color display is used to help people intuitively "grasp" the balance of meaning represented by a set of variables in a complex decision.

Recipient
Keith Lent, Department of Electrical Engineering, University of Texas—Austin
Project Description
A General-Purpose, Flowchart-based Sound Synthesis: Language:
To develop a system that eliminates the awkwardness and complexity of the older sound-synthesis languages, but without removing any of their flexibility. The system will be an icon-based, interactive sound synthesis language, which will allow the user to create and modify digital synthesis processing algorithms by graphically connecting flowchart symbols.

Recipient
Joel Boaz, Department of Anthropology, University of Wisconsin
Project Description
Graphic Display for Archaeological Analysis:
To develop software for the study of the distribution of artifacts across prehistoric archaeological sites. The software will be described in terms of three stages that will be linked into a single unit through the use of HyperCard.

Recipient
Jeffrey T. Spitulnik, Vassar College:
Project Description
Instruction in the Field of Physiological Psychology in a Hypermedia Environment:
A study in the structure and function of the nervous system, complete with an introduction to the discipline and a laboratory simulation including post-lab data manipulation. Psychological theory, methodology, and atlases will be available through HyperCard applications with extensive graphics, using a Macintosh II with a color monitor and various digitizing technologies for graphic image implementation.

Recipients
Hugh Livingston and Igor Popovic, Department of Music, Yale University
Project Description
An Interactive Computer Environment for Music Research and Composition:
Focus will be on addressing in detail the evolution of the Expandable Music Code, the implementation of a symbolic music notation editor, and applications from composition and analysis. Stress will be placed on the graphic representation of facets of music not readily apparent and the use of these for the complementary intertwined processes of analysis and composition.

Products

The **Products** section highlights new or unusual products or uses of existing products that are of interest to the academic community. This issue, we are highlighting new products from Claris and from Apple.

Please send in suggestions about products that you think are worth publicizing to your colleagues in higher education.

New Products from Claris

MacWrite 5.0

Claris Corporation has announced MacWrite 5.0, an enhanced version of the most widely used word processing software program for the Apple Macintosh personal computer.

The new version features a built-in spelling checker with a 100,000-word dictionary, as well as the capability for users to build specialized dictionaries. Other enhancements include keyboard shortcuts for cursor control and for frequently used functions such as inserting rulers, selecting text, and closing, saving, and printing documents and files. MacWrite 5.0 also supports large-screen monitors, allowing full-page display of documents. And the new version continues to provide the easy-to-use interface that was the trademark of the original MacWrite.

The new version features a built-in spelling checker with a 100,000-word dictionary, as well as the capability for users to build specialized dictionaries.

MacWrite 5.0 is compatible with all Macintosh models from the 512K Enhanced through the Macintosh II, with System 3.2 or later and Finder 5.3 or later, and with MultiFinder™ and AppleShare.

MacDraw II

MacDraw II is an all-new version of the Apple Macintosh software tool that set the personal computer industry standard for object-oriented drawing applications.

MacDraw II is three to ten times faster than other drawing programs.

It is a totally rewritten version of the original package, which was the first to bring high-quality object-oriented graphics to average computer users. MacDraw II is three to ten times faster than other drawing programs (including MacDraw). It satisfies user needs in graphic arts, presentations, desktop publishing, and computer-aided design, by providing color capabilities, more flexibility, and more precision.

Other enhancements include libraries of named objects, stored for ready recall; single-letter type font; style and size editing capability; and nine user-defined views within a document, allowing the user to move rapidly from one place to another. In addition, it is possible to customize pen widths, dashed lines, and patterns. A plotter driver is provided for output to selected plotters up to "E" size.

MacDraw II will run on all current Macintosh models and on the discontinued Macintosh 512K Enhanced. It can read and revise all files created with the original MacDraw.

MacPaint 2.0

Rejuvenating the original painting program for personal computers, Claris Corporation has announced MacPaint 2.0, a powerful, free-form graphics tool for the Apple Macintosh personal computer. The new version features multiple windows, "tear-off" tool palettes, and a "Snapshot" function for selective retouching.

MacPaint 2.0 retains the ease of use associated with the original program, while offering significant enhancements.

MacPaint, the very first application ever written for the Macintosh, is the most popular graphics program for the machine. It has been widely used both by design professionals who require free-form sketches or special effects and by novices who need presentation-quality business graphics quickly and easily. MacPaint 2.0 retains the ease of use associated with the original program, while offering significant enhancements.

The "tear-off" menus allow users to take advantage of large Macintosh monitors by locating tool and pattern palettes anywhere on the screen, while users of the standard Macintosh screen may opt to keep one or both palettes hidden under the menu bar. Another benefit for small-screen users is automatic scrolling of drawing and selection tools. An innovative new "Snapshot" feature lets users return any portion of a drawing to the way it appeared in a previous version. Used in conjunction with the "magic eraser," Snapshot can be a powerful retouching tool. To allow you to edit drawings more accurately and efficiently, MacPaint 2.0 now lets you use all the drawing tools when viewing a document at 50, 200, 400, or 800 percent.

MacPaint 2.0 is compatible with the Macintosh 512K Enhanced, Macintosh Plus, Macintosh SE, and Macintosh II models.

MacProject II

MacProject II is a significantly enhanced version of its best-selling project-management program MacProject. Addressing the needs of project-management professionals, MacProject II provides powerful capabilities, while retaining the easy-to-use graphic interface of the original program.

MacProject II is designed for project managers, planning consultants, architects, engineers, and construction planners who must efficiently manage projects, regardless of their size. The program's intuitive interface enables users to concentrate their efforts on project deadlines, budgets, and workforce availability.

MacProject II includes many new features that are optimized for managing intricate projects. These include the ability to manage an unlimited number of tasks per project. Subprojects can now be linked and consolidated with other projects. Multiple resource calendars (up to eight) allow users to customize work calendars according to resource availabilities. Planned and actual task status are displayed in the Schedule Chart, Gantt charts, and in the Project Table. And Resource Histograms allow planners to graphically display resource usage.

MacProject II includes many new features that are optimized for managing intricate projects.

MacProject II is compatible with the Macintosh 512K Enhanced, Macintosh Plus, Macintosh SE, or Macintosh II computer.

SmartForm Designer and SmartForm Manager

SmartForm Designer and SmartForm Manager are two software programs for the Apple Macintosh computer that enable the creation and management of high-quality forms of all types, such as expense reports, invoices, purchase requisitions, and loan applications. Forms can be printed with high-resolution printers for manual completion or distributed electronically for on-line completion.

SmartForm Designer is an object-oriented drawing program optimized for forms creation. It includes forms-specific graphics tools, as well as powerful data-definition features that allow for electronic capture and the subsequent manipulation of data. SmartForm Manager allows for the quick completion of electronic forms created with SmartForm Designer. The program provides intelligent assistance to those users who are filling out forms on screen, including context-sensitive help, calculations, error checking, and automatic formatting and completion of entries. The program also allows users to manage their electronic forms through sorting and searching capabilities, and through a flexible interface with other software programs such as databases.

SmartForm Designer and SmartForm Manager provide the capability to do everything with forms—create, print, distribute, and capture data. These programs simplify work with forms by allowing companies to control the forms creation process. Because they're electronic, forms can be changed easily, printed on demand, and distributed instantly. Information can also be shared with an external database or spreadsheet.

For more information about Claris products, or the name of the Claris dealer nearest you, call 800-334-3535. For information about upgrade programs, call 800-544-8554.

New Products from Apple

Apple Announces a CD-ROM

Apple Computer has announced the AppleCD SC™, a CD-ROM (compact disc, read-only memory) drive to be used with individual Macintosh and Apple II computers or shared by users connected to the AppleShare file server.

CD-ROMs can store the resources of a small library, providing users with access to vast quantities of information—including text, digitized images, and sounds. A single disc holds as much as 700 double-sided Macintosh floppy disks—well over 550 megabytes, or about 270,000 pages of typewritten text.

To expedite development of CD-ROM products, Apple has designed its drive to be compatible with the International Standards Organization's standard file format, commonly known as "High Sierra." Macintosh and Apple II file support will be available when drives are first shipped to customers, and ISO/High Sierra support will become available in early summer. Customers who purchase drives before this time will receive a software upgrade free of charge.

Apple also plans to support developers by offering a starter kit that will give them a cost-effective way of quickly developing prototype CD-ROM discs. Specific elements of the starter kit will be announced to developers before the AppleCD SC ships.

As an ongoing service to interested developers, Apple will demonstrate how its HyperCard software can be used with the CD-ROM drive. In addition to its other capabilities, HyperCard is an open-ended development environment that can be customized for particular projects involving CD-ROM. For example, the Project Jefferson team used the extension capabilities of HyperCard, adding functions that are important when searching very large relational databases. Apple's technical staff will work closely with publishers in augmenting HyperCard functions.

As an ongoing service to interested developers, Apple will demonstrate how its HyperCard software can be used with the CD-ROM drive.

UNIX Power with a Macintosh Personality

Apple Computer is now shipping A/UX, its implementation of the AT&T UNIX® operating system for the Macintosh II computer, which combines the intuitive user interface of the Macintosh with the power of UNIX.

UNIX is a powerful computer operating system that's highly popular among technical, computer-sophisticated users. A/UX preserves the power of UNIX and adds Macintosh ease of use, bringing the benefits of UNIX to more customers and enhancing the Macintosh II computer's utility as a full-function workstation. With A/UX, the Macintosh II can meet many of your institution's research and instructional needs, especially within computer science and engineering departments. And A/UX provides direct TCP/IP connection to your campus network.

With A/UX, the Macintosh II can meet many of your institution's research and instructional needs.

Based on the most popular UNIX standards, including a full implementation of AT&T System V.2.2 with Berkeley (BSD) 4.2 and 4.3 extensions, A/UX provides easy portability of existing UNIX applications, as well as networking. It has passed all tests of the System V Validation Suite (SVVS), and conforms to the System V Interface Definition (SVID).

Already, more than 25 developers have announced more than 50 A/UX-compatible products, ranging from relational databases and office-automation applications to networking tools, languages and compilers, large-screen displays, and tape drives.

Applications written for A/UX can take advantage of the Macintosh Toolbox features to create the standard Macintosh user interface. Some of the applications currently available have incorporated the Macintosh user interface; others are expected to switch to it in future revisions.

A/UX provides all the capabilities of UNIX—such as virtual memory, multitasking, interprocess communication, and shared networking and relational databases. Designed for ease of use in both its user interface and system administration, A/UX lets new groups of nontechnical users reap the benefits of UNIX.

Development Environments
In addition to using the Macintosh Toolbox, as mentioned earlier, A/UX programmers will be able to develop on the X-Window System, an emerging industry standard for network-based windowing, offering portability among hardware platforms.

C, Bourne, and Korn shells are also included, as are Documenter's Workbench (DWB), Programmer's Workbench (PWB), and Adobe Systems' TranScript to provide PostScript output on Apple's LaserWriter products from UNIX utilities such as DWB.

A/UX networking capabilities add to Macintosh connectivity, allowing groups using a variety of equipment to perform file transfer, remote log-in, sharing of peripherals and servers, and sophisticated file-sharing functions.

To help users get the most out of A/UX, Apple is offering a comprehensive, ongoing support program.

Product Support
To help users get the most out of A/UX, Apple is offering a comprehensive, ongoing support program, which includes services that provide regular software and manual updates. Customers who pay for these update services also can subscribe to the A/UX Hotline, an option that puts users in close touch with technical experts in case of difficulties.

A/UX Q&A

The following is a Q&A with Apple's Bill Jacobs, product manager for A/UX.

Q. Why does a Macintosh II with A/UX cost so much more than a low-end Sun workstation?

A. It doesn't. First, our base configuration includes a disk—the low end of the Sun, Apollo, HP, and DEC product lines are diskless systems. Further, the low-end products from those companies are restricted configurations. The Macintosh II offers six industry-standard NuBus expansion slots; the other low-end products I mentioned do not.

Q. How does the performance of a Macintosh II running A/UX compare to the competition?

A. The Macintosh II running A/UX peforms very close to the Sun 3/50 disk-based workstation, and is faster than diskless versions of similar products.

Q. What is Apple doing about X/Windows? What version will you support?

A. Apple will port X.11 Version 2 to Macintosh II under A/UX, and make it available to Apple customers.

Q. Will Apple implement the POSIX standard for UNIX systems?

A. Apple is investigating POSIX now. We are committed to supporting industry standards as they emerge.

Q. Can I develop software to run on A/UX and the Macintosh operating system?

A. Yes. You can develop programs that will compile easily under Macintosh operating system and under UNIX. They will use the same toolbox calls and have the same user interface look and feel.

Q. Can I run Macintosh applications under A/UX?

A. A/UX offers the Macintosh Toolbox. For developers who substantially follow the rules for programming Macintoshes set forth in *Inside Macintosh,* the same programs can be run under A/UX. We expect lots of Macintosh operating-system developers to label their software for A/UX. These developers are responsible for verifying that their programs function under A/UX. We've found that while only a few Macintosh applications run without change, a very large percentage will run with minimal changes. Our goal is to make this attractive and simple for Macintosh developers.

We expect lots of Macintosh operating-system developers to label their software for A/UX.

Q. What Macintosh applications run under A/UX?

A. The list isn't completed yet, because most of the developers don't yet have access to A/UX systems. We think that a large share of Macintosh developers will verify their applications under A/UX.

Q. Does HyperCard run under A/UX?

A. Not yet. Since it's an integrated environment in itself, we don't see lots of switching in and out of HyperCard from A/UX. However, Apple will ensure that future revisions of HyperCard can run under A/UX.

Q. Does A/UX support AppleTalk?

A. We have not yet implemeted AppleTalk under A/UX. Watch for it soon.

A/UX Products for Higher Education
Database products:

INFORMIX-SQL
INFORMIX-ESQL/COBOL
INFORMIX-ESQL/C
INFORMIX-ESQL/Ada
INFORMIX C-ISAM
INFORMIX-Rapid Development
System: Interactive Debugger and Interpreter
Contact: Marina Donovan, (415) 322-4100

Relational Technology:

INGRES SQL
a 4 GL application development package of tools
INGRES/NET
INGRES/STAR
Contact: Kevin Gallagher, (415) 769-1400

Oracle Corporation
Contact: Kathy Monaco, 800-345-3267

Bear River Institute Courses
A/UX Operating System, Macintosh User Interface for UNIX Programmers, Designing a Macintosh Application
Contact: Tony Meadow, (415) 644-0555

Information Presentation Technology's u(micro)Share
Hardware and software that uses the Macintosh user interface to access UNIX mail, file sharing, virtual disks, and so on
Contact: Olivia Favela, (818) 347-7791

MacNIX
Multiuser Finder™ for A/UX
Contact: Bill Sullivan, (408) 427-0676

New and Enhanced AppleTalk Products

Apple has announced an array of new networking products that reinforce the company's commitment to expanding and enhancing the AppleTalk Network System. The new and enhanced products offer more convenient network administration and—for the first time—give users of the enhanced Apple IIe and the Apple IIGS full access to the AppleTalk network and AppleShare file server.

Apple has announced an array of new networking products.

The addition of Apple II support extends the AppleTalk Network System to users in primary through secondary education, where there is a large installed base of Apple II computers. (The network currently supports Macintosh, MS-DOS, and Digital Equipment Corporation's VAX computers.)

A brief outline of the new products:

Apple II Workstation Card and AppleShare IIGS Workstation Software

The Apple II Workstation Card for the enhanced Apple IIe and the AppleShare IIGS Workstation Software for the Apple IIGS computer let users connect to an AppleTalk network via LocalTalk™ cabling for shared access to an AppleShare file server, as well as to networked LaserWriter and ImageWriter printers. Users have access to all the features of the AppleShare file server, including "AppleShare-aware" ProDOS® applications and file and print service capabilities. One Apple IIGS Workstation Software package is required for each file server and one Apple II Workstation Card is required for each enhanced Apple IIe computer.

AppleShare File Server, Version 2.0

AppleShare 2.0 provides several new features that will especially benefit higher education users. These features improve server administration and startup time, along with security of data and applications. They include the following:

AppleShare 2.0 provides several new features that will especially benefit higher education users.

- Foreground server administration—This means that the server remains available for use during server administration activities.
- Copy-protection setting for applications—This allows the network administrator to set application programs launched from the server so that they cannot be copied by users on the network.
- User control of passwords—This lets users change their own passwords.
- CD-ROM volume support—This allows information on CD-ROM drives attached to an AppleShare file server to be shared by network users.
- Support for Apple IIGS and Apple IIe workstations.

AppleShare Print Server

The AppleShare Print Server, an enhanced version of LaserShare™ print spooling, now offers concurrent support of up to five printers. A single server can handle combinations of networked LaserWriter, ImageWriter II, and ImageWriter LQ printers. The AppleShare Print Server software runs concurrently with the AppleShare File Server software on the same Macintosh, letting users continue working instead of waiting for documents to print. Other features include centralizing print-queue management and letting network administrators print high-priority documents immediately, remove documents from the queue, or suspend printing. The AppleShare Print Server can be accessed from any Macintosh, enhanced Apple IIe, Apple IIGS, or MS-DOS workstation on the network.

Inter•Poll Network Administrator's Utility

The Inter•Poll™ Network Administrator's Utility is a software application that contains tools an AppleTalk network administrator can use to implement a network support program. Inter•Poll allows the network administrator to determine the versions of system software running on the network and to test the integrity of the network links. It includes an administrator's guide with suggestions for network planning and configuration; tips for network maintenance and troubleshooting; a set of network mapping tools; and the AppleTalk Responder, a Macintosh resource that lets Inter•Poll obtain information about Macintosh workstations.

Aristotle

Network administrators or teachers can customize student menus and easily modify available selections at any time.

Specifically designed for primary and high schools, Aristotle™ gives students, teachers, and others using individual enhanced Apple IIe and Apple IIGS workstations fast and easy access to ProDOS applications stored on an AppleShare file server. The Aristotle application selector resides on the server and can be shared by all enhanced Apple IIe and Apple IIGS workstation users. Aristotle also features a menu-management program that lets teachers and network administrators specify which applications each study group can access. Network administrators or teachers can customize student menus and easily modify available selections at any time.

AppleTalk for VMS 2.0

AppleTalk for VMS is a set of developer's tools that provide AppleTalk software that runs as a process under DEC's VAX/VMS operating system. AppleTalk for VMS allows a VAX computer to connect to Macintosh and other computers on the AppleTalk network. Applications built on top of AppleTalk for VMS may include:

- File service
- Terminal services
- Print spooling
- Access to DEC printers
- AppleTalk/DECnet gateways
- Cooperative database applications

Connecting to IBM Networks

MacAPPC™ is a set of developer's tools that provide connectivity to IBM-based networks. It gives programmers the software tools to develop communications services between Macintosh and IBM Systems Network Architecture (SNA) networks. The program includes the SNA Logical Unit 6.2, which allows the development of Macintosh applications that dynamically exchange information with IBM-based applications. The MacAPPC server code resides on a NuBus card that fits into an expansion slot on the Macintosh II.

MacWorkstation

MacWorkstation™ is a Macintosh application that allows programmers to use a host computer to communicate with a Macintosh through the standard Macintosh interface, without their having to learn the details of Macintosh programming. It lets VARs, MIS departments, and systems integrators take advantage of Macintosh windows, pull-down menus, and dialog boxes within their own host-based applications, from their Macintosh computers. The MacWorkstation program can be used with a Macintosh 512K Enhanced, Macintosh Plus, Macintosh SE, or Macintosh II computer.

MacWorkstation includes:

- Message Protocols—Messages sent to or received from a host.
- Directors—Elements of a high-level toolbox, such as Alert and Dialog.
- Exec Modules—Macintosh code resource.
- Communications Connection Language (CCL)—Scripting language for accessing remote locations.
- Communications Modules—Macintosh code segments that allow Macintosh to work with different communications protocols.

For more information on Apple products, contact your authorized Apple reseller or Apple education representative.

Keeping Track of Student Grades

GradeBook is an application that allows educators to record, edit, average, and plot grades easily.

GradeBook is an application that allows educators to record, edit, average, and plot grades easily. The screen display looks like a paper grade book, including dog-eared pages and a spiral binding. To turn the page, teachers click on the dog-ear. To edit a student's name or grade, they just point to its cell and type; there are no separate editing windows or dialog boxes. GradeBook performs all calculations "in the background," so averages are always up-to-date. To top it off, a single click instantly shows a trend line or histogram.

GradeBook provides Clipboard support and full Undo capability. It runs on any Macintosh computer system with at least 512K of memory and 800K of disk space. For more information, contact Bobbing Software, 67 Country Oaks Drive, Buda, TX 78610; (512) 295-5045.

New Bibliographic Tools

Personal Bibliographic Software, Inc. announced the interface compatibility of bibliographic information management and retrieval products, Pro-Cite for the Macintosh and MEDLINE Knowledge Finder.

Aries Systems' Knowledge Finder is a probabilistic bibliographic retrieval system that simplifies end-user searching of MEDLINE, while retaining the power and flexibility offered by traditional retrieval systems. MEDLINE citations retrieved from the Knowledge Finder CD-ROM database can be effortlessly exported to PBS's personal reference management tool, Pro-Cite for the Macintosh, which provides advanced facilities for creating and manipulating personal bibliographies and reference databases. Both Pro-Cite and Knowledge Finder software run on the Apple Macintosh computer. For more information, contact Personal Bibliographical Software, Inc., P.O. Box 4250, Ann Arbor, MI 48106; (313) 996-1580.

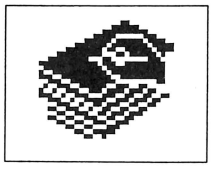
LaserCount

LaserCount is a software system that keeps track of the number of pages printed on your LaserWriter or Linotronic. It also tracks the amount of time that individual Macintosh computers are used on the network with the printer. Access to the Macintosh and the printer is controlled by a log-on and log-off process that generates a billing receipt when users are finished. With LaserCount, you can track usage on an account-by-account basis and collect operating statistics as often as you like. Two programs are used to put the system in to place: LaserCount Administrator and LaserCount.

LaserCount is a software system that keeps track of the number of pages printed on your LaserWriter or Linotronic.

To use LaserCount, you must have a Macintosh with at least 512K of RAM connected via LocalTalk cabling to an Apple LaserWriter, LaserWriter Plus, or Linotronic 100 series printer. A dedicated Macintosh is not required. For additional information, contact LaserCount Systems, Inc., 103 Blue Ridge Trail, Austin, TX 78746; (512) 327-2778.

HyperCard



The **HyperCard** section covers some of the development—and captures some of the excitement—that HyperCard is causing on campuses around the country. It is divided into two parts: Scripts and Tips, and Stacks.

If you have any insights into the workings of HyperCard, or have written a stack that you consider to be of interest to others in the university community, please send us your information.

HyperCard Scripts and Tips

Windoid is a publication for the informed HyperCard user and the newsletter for the Apple HyperCard User Group. Our sincere thanks to the editor and his staff for permission to extract various highlights for our readers. For more information about *Windoid*, send a self-addressed stamped enveloped to the editor, David Leffler, at the following address:

AHUG c/o David Leffler
Apple Computer, Inc.
M/S 27AQ
20525 Mariani Ave.
Cupertino, CA 95014

HyperCard 1.1 Update

HyperCard 1.1 contains many user-requested fixes and enhancements to make your stack use and creation just that much more elegant. It updates your Home card with the new Text Arrows check box and some new Import and Export buttons. Additionally, there are new and improved internationalized Phone, Area Codes, and Datebook stacks. The Text Arrows check box allows you to choose whether you want to use cursor keys in text fields. With the box checked, you can use your cursor keys to move about in text fields. If you then want to go from card to card, just press the Option key first. Naturally, if the box is not checked, everything works just the way you are used to.

HyperCard 1.1 contains many user-requested fixes and enhancements to make your stack use and creation just that much more elegant.

14 HyperCard User Tips

By Phil Wyman

1. Tear off the Tools menu. If you double-click on the Eraser tool, you'll erase all card paint—which can be quite shocking if you do it by mistake. If this happens to you, select Undo from the Edit menu, press the tilde (~) key (the Escape key) or select Command-Z from the keyboard before doing anything else. This will bring back your paint. If Command-Z doesn't get it back, you can try Revert from the Paint menu. Revert takes you back to the state the paint was in when you last entered the card. Leaving a card, even for a moment, is the same as saving all the paint elements on that card or choosing Keep from the Paint menu.

2. To initialize a variable in HyperTalk™, you should put the assignment of the variable in an "on openStack" handler. You will normally only want to initialize your variables the first time you enter your stack. If this is so, you will need to put the following into your openStack handler:

   ```
   on openstack
       global var
       if var is empty then put 12 into var
   end openstack
   ```

   This way, no matter how many times the user comes into the stack during a HyperCard session, the variable "var" will only be assigned this value the first time, since the second time into the stack the global variable will no longer be empty.

3. If your hard disk is not an SCSI drive—or, heaven forbid, you have a floppy-based system—and you have tried to play a long, sophisticated sound in HyperCard, you might notice very poor sound quality, with numerous clicks and pops. These are caused by your disks access time being too slow. There is, however, a nice workaround (as suggested by Barbara at Farallon Comput ing). To stop HyperCard from accessing the disk (doing any processing) while the sound is playing, try this in your script.:

   ```
   play "nameOfSound"
   wait until the sound is "done"
   ```

4. A tip from Paul Foraker: If the message box is already open, and you are typing in a field, to start typing in the message box without moving your fingers from the keyboard, just press Command-M twice.

5. Tear off the Tools windoid. Double click on "A" to access different fonts and font attributes for typing Paint text onto the card or background. If you have already started typing and have

not clicked anywhere else, when you change the font, you will change everything you just typed. If you have already started typing and you wish to change the font only for what you type next, then click with the mouse at the point where you want to start, before again double-clicking on the "A" in the Tool menu.

6. "Not" in HyperTalk is similar to "not" in English. Therefore, you can say:

"Not" in HyperTalk is similar to "not" in English.

```
if variable is not "Yes" then go next card
```

You may also think of "Not" in HyperTalk as a Boolean form of the false condition. For example, you could use the following script if you have a button that you want to alternately hide and show a field...

```
on mouseup
  set visible of field "fldName" to not visible of field "fldName"
end mouseup
```

7. If you protect a stack to allow only the browsing user level, you can reset it to a different level by pressing on the Command key before clicking on the File menu. You will then be able to choose "Protect Stack" and reset the user level.

8. There is a form of the Repeat command that works with "down to." ("Down to" is two words in this syntax.) This form of the repeat command will iterate your variable downward each step through the loop.

```
repeat with x = 100 down to 1
put x into field 1
go next card
end repeat
```

An example of a useful application of the "repeat down to" command is to delete all the buttons or all the fields of a card or background. The following is an example:

```
put the number of buttons into var
repeat with x = var down to 1
choose button tool
click at the loc of button x
doMenu "Cut Button"
end repeat
```

If you had done this with "x = 1 to var" instead of "x = var down to 1," you would soon be clicking at a button that didn't exist.

9. There's a nice debugging feature in the HomeStack script. The handler is called searchscript, and its two parameters are the

There's a nice debugging feature in the HomeStack script.

string you are searching for and the stack that you are going to be searching. What if, for example, you had hundreds of cards with scripts, and you found you had misspelled a word in one script. You could then search all your scripts to find which ones had the same spelling error. You would say in the mes sage box:

searchscript "misspelled word","stackname"

10. You do not have to have your cursor at the end of your mes sage in the message window in order to execute your message. In the message window, you may be typing along on the fifth word and find that you've made a typing error on the first word. You click on the first word and correct your error. Now the cursor is in the first word. At this point you can still press the Return or Enter key to execute the message window, even though your cursor is not at the end of the message.

11. An easy way to copy a current background in your stack into a new background in the same stack is as follows: Copy the card, go to a temporary stack, and paste the card. Immediately copy the card, go to the original stack, and paste the card. This should create a new background in your stack that is exactly like your old one.

12. <the diskspace> is a global property that will tell you how many bytes you have left on the drive that the current stack is located on.

13. When you are printing your script from the script editor, you do not have to print it all. If you want to print just a portion of your script, select the part you want, then click on the print button. Only your selected text will print.

14. Choosing a selection with the Selection tool can leave you with a lot of extra space surrounding your selection. And you may already know that Command-S shrinks a selection rectangle until there is no surrounding space. What if you want a rectangular selection but no extra space? Hold down the Command key before selecting with the Selection tool, and you'll get just that.

HyperCard Utility Scripts

By Robin Shank

The following is a small collection of utility scripts that I find useful in my work.

Finding the Finder. When running under MultiFinder, the following script will hide the HyperCard window, the message window, and any tear-off windoids, and put you in the Finder. It is handy when installed into a button on the Home card, especially when you are working on a small-screen Macintosh. When you choose HyperCard from the Apple menu, the HyperCard windows will reappear exactly as you left them.

Note: Hide/Show card window is a new feature of version 1.1, so you must be running that version for this script to work.

```
on mouseUp
  hide card window
  doMenu "finder"
end mouseUp
```

The following is another handy script that uses hide/show card window. This utility allows you to toggle between settings for small- and large-screen Macintosh computers. One useful example is to put the word "setWindow" under "StartUp" in your Home stack script and put the following script at the end. When you first open HyperCard, you can have your windows right where you want them.

```
on setWindow
  answer "Which CPU?" with "Mac II" or "Mac +"
  if it is "Mac II" then
    show menuBar
    show card window at 20,50
    show tool window at 524,198
    show message window at 0,370
  else if it is "Mac +" then
    show card window at 0,0
    hide tool window
    hide pattern window
    show msg at 30,260
    hide msg
  end if
end setWindow
```

Recovering lost buttons. Well, this script won't recover anything you've already lost, but it may save you a headache in the future. Put this script in the Home stack script. Whenever you try to get rid of a button, this script will complain and give you the chance to get your button back.

```
on deleteButton
  answer "Do you REALLY want to delete that button?" ¬
  with "No, Copy it" or "Yes"
  if it is "No, Copy it" then
    doMenu "copy button"
    doMenu "paste Button"
  end if
end DeleteButton
```

Changing a button's font style. Yes, there are alternatives to those generic buttons with text in good old Chicago. Buttons can be assigned a font and font style through a script. If you wish to change the font of a single button, just type set the textFont of card button "fred" to "Geneva" into the message box, and press Return. In addition, you can set the size (set textSize...), style (set textStyle...) alignment (set TextAlign...), and height (set textHeight...), of any button. My current personal favorite is a shadowed button with 10-point Geneva text.

Once you get the hang of changing the text-related attributes of buttons, it can get a bit tedious to set all these styles by typing lines into the message box. The following script installed into a functionKey and invoked from the message box can be a real finger saver. It will change the button you specify to your choice of font, font size, and font style.

```
on ChangeButton
  ask "Which button?" with "Card button ID"  & id of last button
  if it is empty then exit ChangeButton
  put it into btnname

  ask "Font, Size, Style?" with "Geneva,9,bold"
  if it is empty then exit ChangeButton

  if first item of it is not empty
  then do "set textFont of"  & btnname & " to " & first item of it

  if item 2 of it is not empty
  then do "set textSize of"  & btnname & "to" & item 2 of it

  if item 3 of it is not empty
  then do "set textStyle of" & btnname & "to" & item 3 of it

end changeButton
```

Invoking this script will present two dialogs. In the first dialog, enter the full name of any button on that card. Remember to specify card or background button, and refer to it by name, ID, or number (background button "New button"). In the second dialog, enter the font, size, and style, separated by commas. If you do not wish to change one of the attributes, leave it blank (font, ,style) or enter its current setting.

An alternative solution. Locked fields can behave just like buttons. Create a field, enter your title, choose your text style through the Textstyle window, then lock the field. You can now put any "mouse" handler (on mouseUp, and so on) into the script of this field, and it will behave just like a rect button, with the exception of highlighting.

Which line did I click on? The following function scripts return which line of a field was clicked on. In this case, "line" is defined as the number of textHeight increments from the top of the field, and not as a line of text ending with a carriage return. Also, the field needs to be locked when you click on it, or you will end up putting an insertion point in the field, not executing a script.

Here's a simple example of how this might be useful. Put the following handler, as well as one of the functions, into the script of a field. The field should be locked and contain some lines of text that do not wrap.

```
on mouseUp
  answer "You clicked on line" & quote & line lineclicked()¬
  of card field "newt" & quote
end mouseUp

function lineClicked
  return ((the mouseV - item 2 of the rect of the target) ¬
  div the textHeight of the target) + 1
end lineClicked
```

If you wish this to work on a scrolling field, it becomes a bit more complex to account for the possibility of lines scrolled off the top.

```
function lineClicked
  return (round((the scroll of the target / the textHeight¬
  of the target) ¬
  +  (((the mouseV - item 2 of the rect of the target) ¬
  div the textHeight of the target) + 1)))
end lineClicked
```

Auto-compaction will keep your stacks trim. Install this script into your Home stack script. If any stack you are about to leave needs to be compacted, it will let you know. This is an easy way to ensure that you always have lots of room on your hard disk (...for more stacks).

```
on closeStack
  if the freeSize of this stack >= 15000 then
    answer "This stack has" & round(the freeSize of this¬
      stack/1024) & "k free, Compact?" with  "NO" or "OK"
    if it is "OK" then doMenu "compact stack"
  end if
end closeStack
```

Scrolling Multiple Fields

By Bill Champ

How can you get several scrollable fields to scroll together?

Put these two handlers in your card script.

```
On openCard
  global initScroll
  put 0 into initScroll
  repeat with i = 2 to 5
    set the scroll of card field i to initScroll
  end repeat
end openCard
```

That script simply sets all the scrollable fields you want (in this case, fields 2 thru 5) to zero—scroll

box at the top of the scroll bar. It also stores that scroll value (zero) in a global variable (initScroll).

```
on updateScroll
  global initScroll
  repeat with i = 2 to 5
    set the scroll of card field i to initScroll
  end repeat
end updateScroll
```

This is the workhorse script. It sets the scroll of the fields you want, all equal to the variable initScroll. initScroll gets changed by the user scrolling one of the fields. That's what the next script does.

Put the following script in each of the scrollable fields:

```
on mouseWithin
  global initScroll
  if the scroll of me <> initScroll then
    get the scroll of me
    put it into initScroll
    updateScroll
  end if
end mouseWithin
```

This script detects changes in a field's scroll bar and changes the value of the global variable initScroll. Then it calls the updateScroll script (on the card) to change all appropriate fields to the same value.

These scripts are great for creating tabular style fields that can scroll and yet keep all the information lined up between fields. It helps to lay out the fields evenly horizontal on the card.

Controlling User Inactivity

If you want to provide a way for a stack to regain control after a specified period of inactivity by the user, the following should do it.

Put these three handlers into the Stack script.

```
on idle
  global IdleTime
  if the ticks > 1200 then Welcome
  pass idle
end idle
```

```
on Welcome
  global IdleTime
  repeat until the mouse is down
    -- do whatever the stack does while waiting for a "customer"
  end repeat
  -- MAKE SURE you update IdleTime BEFORE the script ends!!!
  put the ticks into IdleTime
end Welcome

on openStack
  global IdleTime
  put the ticks into IdleTime
end openStack
```

Then put something like the following in EVERY SCRIPT that responds to user actions (scripts that show that a user is not IDLE).

```
on mouseUp
  global IdleTime
  -- do whatever is supposed to be done here
  --
  -- MAKE SURE you update IdleTime as often as the user is not idle.
  put the ticks into IdleTime
  pass mouseUp
end mouseUp
```

Essentially, the "on idle" keeps checking the difference between the current time and the time in IdleTime. If the difference ever exceeds 20 seconds (1,200 ticks) then the Welcome routine is re-started. (You may want to consider changing the 20 seconds to 50 or 60.)

For this to work, every time the user is doing something, the IdleTime variable needs to have the current time stuffed into it. Do it at the END of scripts so the time is more current. Make sure that ANY time the user is active, the IdleTime variable gets updated before that script ends.

If you get stuck or something, feel free to contact George W.J. Shearer, Biola University, Math & Computer Science, 13800 Biola Avenue, La Mirada, CA 90639; (213) 944-035, ext. 3735 or AppleLink: U0235.

Sort a HyperCard Field

The following script has been used for two months, and it is the clearest expression of the algorithm that I have ever seen. Comment out the lockScreen commands to watch it. (Too bad that it is so slow—one of my first XCMD projects will be to write a fast sort for fields.)

I also wrote a recursive QuickSort, but it fails with the "too many variables" message trying to sort just 18 lines. Does this mean that HyperTalk has serious memory limitations?

To sort words, change "line" and "lines" to "word" and "words."

To sort by, for example, the second word of a line, change "it >= line i" to "word 2 of it >= word 2 of line i."

```
On mouseUp -- Shell's sort for HyperCard fields.
  Set lockScreen to true -- (Comment out to see sort in action.)
  Set cursor to 4 -- Wristwatch
  Put number of lines of field 1 into N
  Put 1 into m
  Repeat while m < N
    Put 3*m+1 into m
  end repeat
  Repeat while m > 1
    Put m div 3 into m
    Repeat with j = m + 1 to N
      Get line j of field 1
      Put (j-m) into i
      Repeat until i <= 0
        If it >= line i of field 1 then exit repeat
        Put line i of field 1 into line (i+m) of field 1
        Subtract m from i
      end repeat
      Put it into line (i+m) of field 1
    end repeat
  end repeat
  Set lockScreen to false -- (Comment out to see sort in action.)
  Set cursor to 5 -- Arrow or browse tool
end mouseUp
```

Lost and Found

By Paul Foraker

In the process of designing your own stacks, once in a while you're likely to forget things. If you've ever hidden a field or a button from yourself, you might appreciate this little script.

```
-- make a new button anywhere on your card
-- (you'll delete it later) and put this script in it:
on mouseUp
  repeat with i = 1 to the number of fields
    -- or background buttons, or card fields, or buttons
  show field i
    -- or background button i, etc.
  end repeat
end mouseUp
```

Now, when you click on this temporary button, all your hidden fields or buttons will come into view.

Setting Scripts of Objects on the Fly

There are times when you want a HyperTalk script to make new objects (like buttons or fields) and put scripts into them. The command for getting scripts into objects is set script of <object> to <source>. This command can be used in a number of different ways. Let's look at some examples. (For the sake of simplicity, I've assumed that we're using a simple button mouseUp handler to set the script of a button named target.)

```
-- getting the new script directly from the executing handler
-- (remember not to use Option-Return inside a quoted string)
on mouseUp
  set script of button "target" to "on mouseUp" & ¬
  return & "go next card" & return & "end mouseUp" & return
end mouseUp

--getting the new script from a field
on mouseUp
  set script of button "target" to field 1
end mouseUp

--getting the new script from another object
on mouseUp
  set script of button "target" to script of button 2
end mouseUp
```

```
--modifying an existing script
--let's say we think the old script was simply (on mouseUp) "go next
--card", but we're not sure; it might already have a visual effect,
--and we want to add a visual effect:
on mouseUp
   --first check for a visual effect & get out if there is one
  if script of button "target" contains "visual" then exit mouseUp
   -- if not, then modify the script
  put script of button "target" into temp
  put return & "visual effect wipe left" after line 1 of temp
  set script of button "target" to temp
end mouseUp
```

[Thanks to Linda Donovan for the suggestions.]

Cursor Control in HyperCard

You can call your own cursors in any script, but as soon as the script ends, so does your custom cursor's performance. The only way around is to keep the script going until you are done.

For animated cursors:

1. Create the various positions/views of the cursor that are needed.
2. Number them with consecutive numbers.
3. In your script, repeatedly call the consecutive numbers.

```
on WowieCursor
  repeat with CNum = 128 to 134
    set cursor to CNum
  end repeat
end WowieCursor
```

About XCMDs and XFCNs in HyperCard

By Sioux Lacy

Tips and Techniques for Users of External Code Resources

Many of you may not be aware of yet another tool HyperCard has provided for the stack designer/user. External commands or functions can be used to create an extension to HyperCard functionality. Accessed much in the same manner as user-defined commands and functions, XCMDs and XFCNs are code resources that can be attached to any stack.

Several external commands are included with the HyperCard product: "Flash" is an XCMD that will flash the screen (invert black and white, then invert back again). It's attached to the HyperCard application, so it can be accessed from any stack with the command "Flash," followed by the number of times you would like the screen to flash.

If you look in the Documents Representative stack, you will find another example of an external command: "getdocs." It is called by the Update button's script. A line in the "mouseUp" handler reads:

```
getdocs line i of documents
```

This is a call to the "getdocs" command, passing it one argument that is a folder name from the Documents card in your Home stack. "Getdocs" responds by creating the cards in the Documents stack.

Already some interesting external commands have appeared as shareware. Some of them provide the user with a button that will attach the external code to your stack. If you discover one that doesn't, you will need to use a resource mover, like ResEdit, to copy it from its current stack, and paste it into your stack. In ResEdit, this procedure works as follows:

1. Launch ResEdit.
2. Find the stack with the external command or function that you want to copy, and open that stack.
3. You will see a resource type of XCMD (or XFCN).
4. Select and copy the resource.
5. Then find and open your stack.
6. It will probably ask you if you want to open a "resource fork" for your stack.
7. Click OK, and ResEdit will open a window into which you can paste the resource.

For you Pascal, C, or 68000 assembly-language programmers who want to write your own external commands, here are a couple of hints:

1. Most important, get the HyperCard Developer's Toolkit package from the Apple Programmer's and Developer's Association (APDA), located at 290 SW 43rd Street, Renton, WA 98055. This package includes a disk called Magic Hook, which contains sample source files in Pascal and C, and preliminary documentation.

2. Don't declare any global or static variables. Externals cannot share HyperCard's global data space. You can allocate space on the heap with "NewHandle" or "NewPtr" calls, but be sure to dispose of that space before you return to HyperCard.

3. Some of the callback glue routines also return handles to the caller (specifically, PasToZero, EvalExpr, GetGlobal, GetFieldByName, GetFieldByNum, and GetFieldByID). In these cases, HyperCard has allocated the space associated with these handles, and your code is responsible for deallocating it once you've examined its contents.

4. The rule about disposing of handles and pointers that your code allocates has one exception: if you allocate space in order to return a string to HyperCard in the "returnValue" element of the XCMD parameter block. HyperCard has not preallocated any heapspace for that handle. You must do the allocation in your code, then copy your return message into that allocated space. HyperCard will dispose of the space for you.

An example that combines usage of a callback routine and assignment to a returnValue follows. Note that in this case, you do not deallocate the handle. This example uses the "PasToZero" callback routine to copy a string and zero-terminate it for return to HyperCard.

```
paramPtr^.returnValue := PasToZero('An error has occurred.');
```

5. For testing externals, one useful technique is to call your code repeatedly from HyperCard, and "put the heapspace" in between the calls. If you see a rapidly diminishing heap, you might suspect that your external is allocating space and forgetting to dispose of it.

6. An error in the Magic Hooks documentation: the callback routines, "ExtToStr" and "StrToExt" convert floating-point numbers to strings and vice versa—not extended long integers.

7. HyperCard stores strings as zero-terminated arrays of any length. This is consistent with the way strings are handled in C, but not in Pascal. So, for example, when a Pascal code resource receives parameters, it will need to convert the strings to Pascal strings (beginning with a length byte, and no zero-terminator). Read the documentation for the callback glue routines carefully. Some of those routines return zero-terminated strings, and others return Pascal strings. Your code will need to convert if appropriate.

Also, remember that a Pascal string is limited in length to 255 characters. So, for example, if you have a zero-terminated string that is longer than that, and you call ZeroToPas to convert it for you, HyperCard will truncate your string at 255 characters.

8. If you're writing external code in MPW C, there are some additional constraints that you need to be aware of regarding string literals. In an environment that has global and static data initialization, you can copy a string to the space referenced by a handle in this manner:

```
myHndl = NewHandle( STRINGSIZE );
strcpy( *myHndl, "Hello world" );    /* probably should lock this handle */
                                     /* and unlock it after the strcpy call */
```

In an MPW C (see Lightspeed note below) external code resource, you cannot use a string literal. Instead, you must create an STR# resource that contains your strings (use ResEdit to do this), and then retrieve them with a call to GetIndString:

```
char myString[ STRINGSIZE ];          /* make an array large enough */

GetIndString( myString, RESOURCE_ID,  STRING_NUM );
```

9. It is not necessary to initialize the various Tool Managers in your external code. HyperCard has already done this, and in some cases, such as the GrafPort, reinitialization can be hazardous.

Note: Lightspeed C will allow the string literal to be passed as an argument.

Explanations of Some HyperCard Features

By Robin Shank

Drawing through scripts: Because the Polygon tool is controlled differently from other drawing tools, the Drag From command won't work in this tool. If you need to plot irregular shapes, use the Curve tool, or the Line tool instead. The Lasso tool depends on an enclosing movement to select something, but a script will only drag in a straight line.

In many cases when you draw through a script, you will want to set the dragSpeed down to around 500. With the dragSpeed at the default setting, the Spray tool will only show the starting and ending points, because the spray cycle is slower than the movement on the screen. Also, with Draw Multiple on, you won't see any multiples unless the dragSpeed is slowed down.

Background fields that are overlapped by another object or card picture will print at 72 dpi on the LaserWriter printer. A piece of card paint whose rect overlaps a background field will cause that background field to print at 72 dpi. (The rect can be shown by the rectangle formed by the marching ants when drawn by the selection rectangle.) One exception: if a transparent button that does not show a name or icon covers a background field, that field will still print at high resolution.

After printing a stack or report, the format is saved as the default. However, when printing cards through a script using Open printing with dialog, the dialog will show the default format determined by the last Print Stack, but changes to the printing format are not saved and affect only that one print job.

The absolute minimum memory configuration recommended to run HyperCard under MultiFinder is 750K. If you run HyperCard under MultiFinder on a 1-megabyte machine, you will be limited to Typing UserLevel. Processes that need more memory (such as pasting graphics or printing) will at least complain, and at worst, exit grumpily back to the Finder. (Who said crash??)

The absolute minimum memory configuration recommended to run HyperCard under MultiFinder is 750K.

When printing under MultiFinder with background printing on, Print stack sends copious amounts of data to the LaserWriter in a very short period of time. If a spooler of any kind is catching that before sending it to the printer, it is likely to fill up very quickly. A MultiFinder Spool folder for a 10-card, 20K stack could easily be over 1 megabyte (500K and more is being sent to the LaserWriter per card).

HyperTalk Hints

By James Redfern

A container is what you put your information into. Containers are: "it," "selection," the message box, a field, or a variable. They can be thought of as open or closed. An open container displays information on the screen; the contents of a closed container cannot be viewed. Variables and "it" are closed containers. The number of characters in an open container must be less than 32K, (probably less than 30K to be safe), and less than 5K if a scrolling field is to respond quickly. The number of characters in a closed container is limited by available memory. (An exception is the use of the prepositions "before" or "after" with a closed container, which limits the number of characters to 64K.)

Information is represented as a string of characters. Strings can sometimes be further interpreted as numbers or the logical values: true and false. A number can be further interpreted as an integer or unsigned integer.

When HyperCard performs an operation that requires numbers as operators, it will try to interpret (convert) the strings as numbers.

When HyperCard performs an operation that requires numbers as operators, such as addition, it will try to interpret (convert) the strings as numbers. If the resulting number is put into a closed container, the container will retain the full precision of the internal representation. If the resulting number is put into an open container, the number must be represented (converted) as a string. HyperTalk uses the current value of the numberFormat property to format the string representation.

When HyperCard performs an operation that requires strings as operators, such as concatenation, it will first represent any numbers as strings using numberFormat. The resulting string will remain a string even if it is put into a closed container.

Constants, such as pi or zero, represent strings of digits, and are not interpreted as numbers unless forced to by an operation. Therefore, "put pi" results in "3.141592653..." but "put pi + 0" results in "3.141593" (the "+" forced "pi" to be converted to a number, which had to be converted back to a string since the message box is an open container, so numberFormat was used to round to six decimal places).

Comparison is an operation that will try to interpret strings as numbers. Therefore, "3" > "20" is false, but "*3" > "*20" is true: If the goal is a string comparison and the values could be interpreted as numbers, then a nonnumeric character must be concatenated to both values, such as 'if "*" & variable1 > "*" & variable2 then...' to prevent interpretation as a number. Remember that equality comparisons use only primary ordering of characters, so case and diacriticals are ignored. Magnitude comparisons use both primary and secondary ordering.

Recording Sounds in HyperCard

The MacRecorder Sound System allows Macintosh users to record real sound directly into the Macintosh computer. When used with a built-in or external microphone, any sound source (including television or stereo) can be accessed to make recordings. The compact and easy-to-use system is made up of the MacRecorder sound digitizer, a HyperCard application called HyperSound, and SoundEdit, sophisticated editing software. This is an excellent product. If you use HyperCard, you will really enjoy working with MacRecorder.

Macintosh II users can even record in stereo.

Macintosh II users can even record in stereo, by plugging a MacRecorder into each of the serial ports. Sound can then be played back in stereo on a Macintosh II computer or in mono on a Macintosh SE and earlier Macintosh models. MacRecorder complements the freedom associated with Apple Computer's HyperCard product. Using HyperSound, a utility that accompanies MacRecorder, sound can be recorded from within a HyperCard stack. HyperSound uses a familiar "digital cassette tape graphic" interface to guide users through the application, allowing easy access to sound files. Sound can be recorded, stored, and played back using four sampling rates: the more frequent the rate of sampling, the higher the sound quality reproduced. Another feature of HyperSound is its Copy Sound to Stack button. With this button, any sound can be automatically pasted into any stack. It can be saved and then retrieved when the stack is opened.

SoundEdit is a stand-alone application included with MacRecorder that allows Macintosh users to record, edit, play, and save sounds in a variety of standard sound formats. In addition, it can control sampling rates and provides 8-to-1 sound compression. With SoundEdit's special effects, you can create sounds in loops, mix any sounds, and use the filter like a graphic equalizer.

For more information, contact:
Farallon Computing
2150 Kitteredge Street
Berkeley, CA 94704
(415) 849-2331

HyperCard Books

HyperCard Power: Techniques and Scripts
Carol Kaehler (author of the on-line HyperCard Help stack)
Addison-Wesley Publishing Company
$17.95

The Complete HyperCard Handbook
Danny Goodman
Bantam Books
$29.95

HyperCard Stacks
Stack: Adult Literacy Instruction

One of the activities of the Office of Computing Services of Drexel University is supplying technical support for the Mayor's Commission on Literacy in Philadelphia. In this role, the university provides training and support for some 60 literacy centers throughout the city. These centers have Macintosh, PCjr, Commodore, or Apple II computers for use in adult literacy instruction. We offer on-site training and consulting on the use of software and hardware, and on ways to maximize the use of the computers for instructional and administrative purposes. Since most literacy staff and volunteers have little or no experience in using the computer, this support service is crucial to literacy programs' ability to take full advantage of the available technology.

We have found very little quality instructional software available for adult literacy students.

As we have worked with the various machines, we have found very little quality instructional software available for adult literacy students. Most instructional software, whether for the development of math or language arts skills, is designed for young children and not appropriate for the adult learner. This is particularly true for software in the K–4 grade range. Part of Drexel's support function includes the development of appropriate adult literacy instructional software for the Macintosh—software that addresses basic skills development in a mature (nonpatronizing and noncondescending) manner. In order to produce software in a timely, efficient manner, Drexel has chosen to use HyperCard as its development tool.

Drexel has a long tradition of academic software development for the Macintosh, and currently has nearly 40 programs available through the Academic Courseware Exchange. The university has a full-time programming staff that works with faculty developers to create quality educational software. This allows faculty members to act as the content experts and architects, without having to worry about becoming programmers in order to implement their ideas. The same model is being used with the development of adult literacy software.

The Drexel literacy effort employs one full-time professional and one co-op student programmer. The professional visits the sites and does all the training and consulting for the literacy providers. In addition, part of his time is spent working with adult literacy tutors and coordinators to design appropriate instructional software. The literacy staff members provide the content expertise needed to maintain the quality and relevance of the programs. The co-op student programmer then implements the design concepts and content and does the majority of the coding. In addition, our Software Development Group provides consulting services for the literacy project and tests each program extensively and rigorously.

HyperCard has proved to be an excellent application for this kind of development, for a variety of reasons. First and foremost, pure code does not have to be written in order to create an application. In addition, HyperCard is extremely versatile and flexible, allowing the developer to create a wide variety of programs with one tool. In addition, it allows the developer to integrate voice, text, and graphics effectively to reinforce what is being taught. The program's hypertext capabilities enable programmers to combine various approaches to teaching a skill into one application. With the voice capability, it is also possible to create an application that a student can use without being able to read any directions on—or off—the screen.

The program's hypertext capabilities enable programmers to combine various approaches to teaching a skill into one application.

Several programs have been initiated for the program, and a couple are near completion. One that is complete is the Syllable Sounds program, which opens with a brief introduction to the Macintosh, using synthesized voice, and proceeds to present a variety (23 in all) of two-letter consonant blends and vowel combinations. These may be chosen deliberately or may be generated randomly by the program. Each sound is spelled and sounded out, and a word that uses the combination is presented. In addition, a graphical representation of the word appears. A second version of this program presents another 23 combinations and will be completed this spring.

The second program is an alphabet program for adults. Each letter is presented in alphabetical order, and several words using the letter are spoken and represented graphically. A companion program presents each letter, along with all the possible sounds for that letter. Again, the learning is reinforced with pronunciation and graphics.

These programs are similar to traditional CAI skill practice programs, but other projects are under development that address some comprehension skills using language experience writing projects. One program will allow the user to type in any text and delete selected words or parts of words. The student will complete the selection using contextual clues to fill in the correct words or parts of the words. In addition, related comprehension questions will be asked concerning the written passage.

While software development is an extremely complex, time-consuming activity, using a tool like HyperCard not only makes the task less complex and arduous, but also drastically reduces the time investment required to produce quality programs.

For additional information contact:
Jan Biros
Coordinator, Microcomputing Program
Drexel University
Philadelphia, PA 19104

Stack: Stackware Review

At Michigan State University, we are developing a Stackware Review Stack, which includes a brief description of each stack, its developers, and price structure, and comments on the use of audio and visual material, ease of use, and unusual HyperTalk programming. It is a continually developing stack, growing with each review. We would be happy to share the stack with others, and are interested in receiving copies of stacks to include in the reviews.

For more information, contact:
Dr. Carrie Heeter
Director, Communication Technology Laboratory
Michigan State University
East Lansing, MI 48824-1212

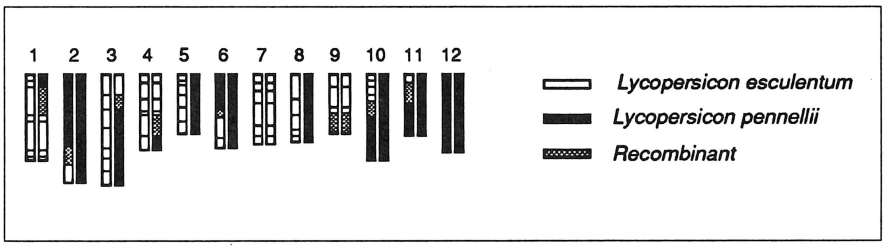
Stack: HyperGene

HyperGene is a Macintosh program to prepare "graphical genotypes." All sexually reproducing organisms (such as humans, other mammals, and higher plants) are "mosaics" of genetic information from their ancestors. Although this fact has been known for many years, it has generally not been practical to determine the genetic constitution for an individual's entire ensemble of chromosomes (genome). This limitation has been due to the paucity of genetic markers in most species, as well as the difficulty in analyzing large numbers of genetic markers simultaneously.

With the advent of high-density genetic maps based on DNA markers, it has become possible to determine an individual's genetic constitution throughout its entire genome. However, because of the large number of data points required to describe such a constitution (which is expressed as a string of hundreds of numbers), it is generally difficult to comprehend or utilize this valuable information.

With the advent of high-density genetic maps based on DNA markers, it has become possible to determine an individual's genetic constitution throughout its entire genome.

HyperGene consists of several HyperCard stacks. The program transforms numerical genetic data into an easy-to-understand graphical form, known as a graphical genotype. An example of a graphical genotype for a hybrid tomato is shown below:



In addition to allowing users to prepare graphical genotypes, HyperGene also contains routines that enable the user to select specific genotypes from a population, analyze the pedigree of an individual by viewing the graphical genotypes of its ancestors, and predict the best breeding strategy to obtain a target genome.

Although HyperGene was developed for use in breeding programs with crop plants, it can also be used to prepare graphical genotypes for humans. Graphical genotypes may be of great value in diagnosing and analyzing complex human genetic diseases.

For more information, contact:
Nevin D. Young and Steven D. Tanksley
Cornell University
Ithaca, NY 14853

Stack: Learn to Read Russian

Learn to Read Russian stacks 1 through 7 are roughly the first half (15 stacks projected) of a Russian course. It features hundreds of digitized Russian words and phrases, so that the user can learn the pronunciation of Russian while learning to read and translate. Its purpose is to serve as a self-study graduate foreign language reading course, the type graduate students take to prepare themselves for a foreign language reading exam as part of their degree requirements.

Russian Vocabulary Stack. This is a utility stack to go with the Learn to Read Russian series. A button in the other stacks will take you to this stack, which lets you look up any word (English to Russian, Russian to English) in the course and then return to where you came from.

Russian Noun Tutor. This stack is a stand-alone application that contains all Russian noun declensions, a tutorial about Russian noun gender, lists of irregularities, and extensive exercises in all cases and numbers. The stack's resource fork contains a Russian font to do the exercises with, so there's no need to have one in your system.

This stack is a port of my Pascal Russian Verbal Aspect with the added benefit that, since it's a HyperCard stack, you can add your own verbs and quiz sentences.

Russian Verbal Aspect. This stack is a port of my Pascal Russian Verbal Aspect, much modified and improved, with the added benefit that, since it's a HyperCard stack, you can add your own verbs and quiz sentences to the existing 20 verbs and 200 sentences. It, too, has a Russian font in the resource fork.

By the way, these stacks are shareware ($7 each). The shareware fee does not have to be sent with the disks and SASE to get the stacks. It is only expected if the stacks are kept and used. If, however, the shareware fee is sent with the request, I'll supply the disks and pay the postage. (That's not a bad deal.)

The stacks can be obtained by sending blank, formatted 800K disks (five for the Learn to Read Russian 1 through 7 and Russian Vocabulary stacks, and one each for the Russian Noun Tutor and Russian Verbal Aspect) and a self-addressed stamped envelope to the address given below:

For more information or to order, contact:
Professor Martin P. Rice
Coordinator of Humanities Computing Development
Department of Germanic and Slavic Languages
University of Tennessee
Knoxville, TN 37996

Stack: Branching Logic Case Analysis Template for Instruction and Testing

The stack was developed for instruction in veterinary medicine, but it is appropriate for all disciplines in which case analysis is an important feature.

Developed with HyperCard, this branching-logic case-analysis authoring tool is a template for the development of case simulations. The author builds problems simply by responding to the questions and instructions provided by the program. For example, to develop a case simulation, the author selects the Build Scene button and enters relevant information regarding the case. The author then creates as many as five different options from which the student can select. The options are created simply by clicking on a Build Options button. The template selects the first empty choice, and asks the author a series of questions designed to build the choice. The questions are as follows:

This branching-logic case-analysis authoring tool is a template for the development of case simulations.

Name of the choice
Is the choice "OK," "Poor," or "Dangerous" (this sets the scoring system)
Show message or Go to New Card
Cost of the choice

The responses to the questions create the options, automatically program buttons for each choice, and record the cost and acceptability of each choice. For example, if the author chooses to show a message, a hidden field is revealed and the author is given instructions to enter the message in that field. If the author chooses to create a new card, a new card is produced—which will ultimately contain additional information and choices.

The author creates up to five choices on the first card and then moves on to the next. On each new card, the process is repeated. The end result is that the author has produced a comprehensive simulated case without having to understand HyperTalk. The template "scripts" every button for you according to your responses.

The simulated cases produced can be used for either instruction or testing. The template has a feature that keeps track of the student's score, and automatically calculates and displays a running total of the cost of each choice selected. The template also has a tracking system that monitors each student use, tracks each option selected by each student, and keeps track of both the score and the amount of money spent by each student in each simulation.

For additional information, contact:
Philip A. Bushby
Academic Program Director
College of Veterinary Medicine, Mississippi State University
Drawer V
Mississippi State, MS 39762

Stack: American Architecture

The intent is to enhance the research and instruction of American material culture studies courses.

American Architecture is the first of a proposed three-part HyperCard multistack environment. Its intent is to enhance the research and instruction of American material culture studies courses, via interactive tutorials and interpretive simulations based on a growing visual, documentary, and statistical database gathered through 15 years of artifact research (photographs, maps, site plans, architectural elevations, isometric and schematic drawings, lithographs, charts, and diagrams) and stored on optical media. Each part is based on a chapter of *Everyday Life in Victorian America, 1876-1915* (to be published by Harper and Row in 1989) and can be used in conjunction with the book for course instruction.

This first part, Housing, provides a valuable research tool, while leading students to acquire skills in architectural analysis (techniques for how to research site, form, structure, plan, and style) and cultural analysis (techniques for how to research artistic, social, economic, and cultural change). Students are given "guided tours" through the data to present differing studies and aspects of housing. Later on in the course, they have the opportunity to create and modify their own "tours." One may also test a current hypothesis about middle-class residentiality in late 19th-century America, given a case study. The example here is how the domestic ideal of the Victorian Manse of the 1870s changed to that of the Progressive Bungalow by the 1910s.

American Architecture is based on a more generic complete learning environment comprising several HyperCard stacks. Parts two and three of this project will also be produced from this common base, as could other, similar courseware applications. From the class Home card, the student has access to course information, student rosters, course reading lists, a variety of data stacks, and inquiries that draw upon the database. The data stacks are easily updated and consist of timelines, statistics, pictures, sounds, and animations. There is also a reference set of stacks that contain information for bibliographies, suggested readings, indexes of past student papers and tests, as well as direct links that open textual documents. The course designer creates inquiry stacks that lead the students through various hypotheses based upon the data, and students may switch between related data and the selected inquiry. Students are allowed to create their own inquiries, which are then available for other students to browse. The interface is designed to be tactile and intuitive, with motion and sound cues to help orient the user.

Primary contact:
Thomas J. Schlereth
Department of American Studies
271 Decio Hall
University of Notre Dame
Notre Dame, IN 4655646556-0539
(219) 239-6129
<CDTAXW@IRISHMVS>
AppleLink: A72

Secondary contact:
Mark B. Johnson
User Services
Computing Center
University of Notre Dame
Notre Dame, IN
46556-0539
(219) 239-5600
BITNET: <CDTAXW@IRISHMVS>
AppleLink: A72

Stack: Learn PostScript

Learn PS is a HyperCard application that consists of five stacks—an introduction, a tutorial, an examples library, Help, and Font Fun—that can be used for teaching PostScript programming, as a reference guide to PostScript commands and syntax, or simply for printing programs on a PostScript device.

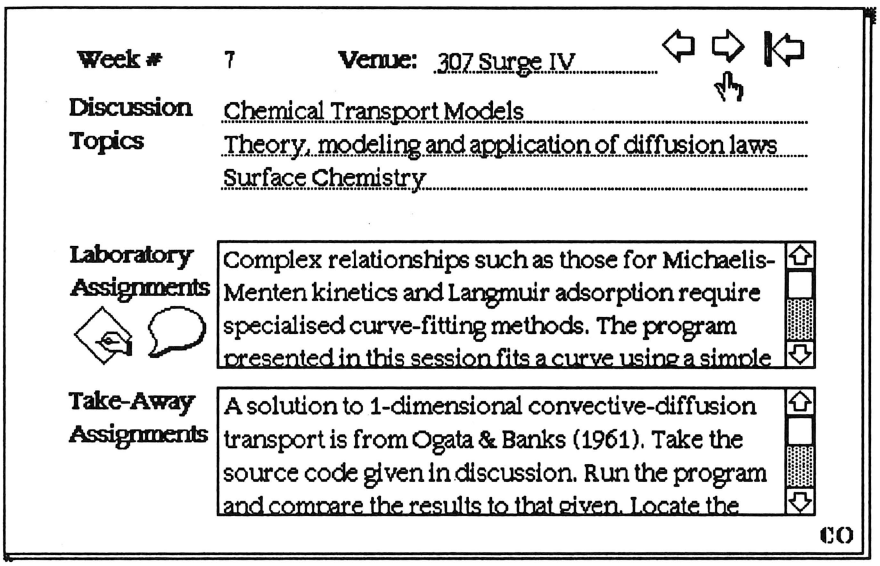
Tutorial PS is a complete lesson in the basics of writing a PostScript program.

The Introduction explains the application's organization and usage and presents an overview of the stacks. Tutorial PS is a complete lesson in the basics of writing a PostScript program. It assumes no programming background and begins with concepts and very simple examples covering the creation of various graphic objects and type. Tutorial PS also provides other published PostScript references. The Examples PS library is a collection of PostScript programs displayed graphically and textually. These can be sorted by level (basic, intermediate, and advanced), searched for the occurrence of any string or command in context, printed or saved to disk, and modified for experimentation or personal use. The Help stack is available from any of the other stacks; it contains a complete listing of all the PostScript commands and their definitions. Commands may be searched for by name, category, or string, and many definitions are cross-referenced with their complementary command. Many commands are also linked to the Examples PS library to provide working examples in context. Finally, Font Fun provides a means for the user to enter the physical attributes of a font—style, value, weight, and proportion—for a desired string and have the program produce the PostScript code necessary to output the string.

Primary contact:
John F. Sherman
Art, Art History & Design
219 Riley Hall
University of Notre Dame
Notre Dame, IN 46556
(219) 239-7175
BITNET: <FRPEXX@IRISHMVS>

Secondary contact:
Mark B. Johnson
User Services
Computing Center
University of Notre Dame
Notre Dame, IN
46556-0539
(219) 239-5600
BITNET: <CDTAXW@IRISHMVS>
AppleLink: A72

Stack: HyperCard-based Chemical Model Laboratory Course



The most recent offering, winter quarter 1988, saw a radical enhancement in the presentation.

For the second time, Water Science 217 (Hydrochemical Models) at the University of California at Davis has presented computer chemical model demonstrations and problems on Macintosh computers. The most recent offering, winter quarter 1988, saw a radical enhancement in the presentation. First, the class was outlined using a HyperCard stack (see figure). The stack included a Home card with buttons linked to cards describing individual weekly assignments, a card for every week in the quarter, a card containing general course information, and several containing additional information pertinent to the assignments. In addition, laboratory handouts were prepared using PageMaker, MacWrite, and Microsoft Excel, all operating under MultiFinder 1.0.

Six out of the eight hydrochemical models were demonstrated on Macintosh computers. Use was also made of one of the campus Macintosh Plus/AppleTalk microcomputing facilities. Programs from previous years written in Microsoft BASIC were ported over by ASCII text transfer. Newly developed programs were written using Turbo Pascal.

The cards shown in the figure include buttons to launch a demonstration program (application icon) and to link to a card with more information card (balloon). The launch feature is extremely useful, because the user is returned to the spot from which the demonstration was launched. Overall, HyperCard is of great benefit in creating a well-structured course layout featuring ease of use and information storage.

By next year, all programs will be on the Macintosh. Development of Macintosh versions of WATEQ and GEOCHEM will begin in late summer.

Several difficulties are inherent in a course such as this. By far the greatest impediment was the reluctance of MS-DOS users to embrace the Macintosh user interface (most despise the mouse!). Furthermore, the most convenient Macintosh available to students in the immediate department was a 512K Enhanced, which would not operate satisfactorily with HyperCard. And students in other departments were provided systems that relied on RAMStart, which gave rise to several moments of anxiety.

Stack: Citation Index

This is a set of HyperCard stacks for cataloging, sorting, and searching through bibliographical citations

This is a set of HyperCard stacks for cataloging, sorting, and searching through bibliographical citations. Searches for single entries or sets of entries can be made on any combination of these criteria: source type (book, article, and so on), author(s), single or multiple subject tags, source name (journal or book title), and date of publication. Entries can be pasted to and from the Clipboard or as text files in MLA, IPA, or Chicago formats.

Citation Index will be distributed on a shareware basis.

For more information, contact:
Terrence Harpold
Comparative Literature and Literary Theory Program
420 Williams Hall
Philadelphia, PA 19104-6305
(215) 898-6836

Stack: Cutural Historical Tour of the Raquette River

Users pursue information on waterpower, recreation, farming, quarrying, water supplies and sewage, ice, transportation, and foraging.

Two undergraduates and I are designing a prototype of an interactive historical exhibit on the cultural ecology and technological history of the Raquette River, which flows through our village. The user "mouses" around in graphics, historical documents, statistics, scientific reports, and maps of the Raquette River in pursuit of an answer to the question, "Why have communities grown up on the banks of this river?" Users pursue information on waterpower, recreation, farming, quarrying, water supplies and sewage, ice, transportation, and foraging. Our final product will be

a prototype of an interactive historical exhibit to display for testing purposes at the village and county museums and among students at Potsdam College.

This tour is built using HyperCard with sounds from McNifty's sound digitizer. Graphics were digitized through Thunderscan 4.0 and MacVision. The prototype stack will be available for trade with other designers of educational programs in May.

For more information, contact:
John T. Omohundro
Professor of Anthropology
Potsdam College
Potsdam, NY 13676
(315) 267-2053

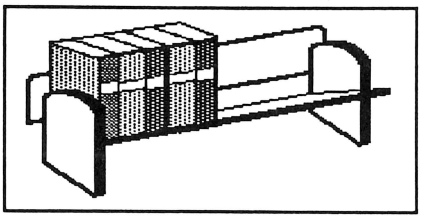
Stack: A Simulated Archaeological Project

We are designing a simulated archaeological project that will allow students to experience the various aspects of archaeology normally not taught in the traditional field school situations. The archaeologist creates hypotheses on the Macintosh, participates in oral grant quest, prepares a budget, decides on a research strategy, and receives excavation data. He or she must then analyze the data (which may include burials, carbon-dating samples, ceramics, and other material remains) after viewing digitized images. The project ends with the submission of a site report. The project is constructed using HyperCard, with images digitized in Thunderscan 4.0 and MacVision. Sounds are digitized in MacNifty and MacinTalk, using an XCMD resource called Talk. This project will be completed in June.

The archaeologist creates hypotheses on the Macintosh, participates in oral grant quest, prepares a budget, decides on a research strategy, and receives excavation data.

For more information, contact:
Steven J. Marqusee, Associate Professor—Anthropology or
Peter Brouwer, Instructor—Computer Information Sciences
Potsdam College
Potsdam, NY 13676
(315) 267-2053

Resources



Authoring

Guide	MACUSER	Jan. 88	p. 127
HyperCard Versatility	MACUSER	Jan. 88	p. 231
VideoWorks II	MACUSER	Mar. 88	p. 150

Business

Easy Checks	MACUSER	Apr. 88	p. 100
Preparing 1987 Taxes	MACWORLD	Mar. 88	p. 148
TaxView Planner	MACWORLD	Mar. 88	p. 172

Computer Science

(BASIC) Banners	NIBBLE	Vol. 3 No. 2	p. 22
(BASIC) Patterns and Cursors	NIBBLE	Vol. 3 No. 2	p. 38
(BASIC) Umbrella Clip	NIBBLE	Vol. 3 No. 2	p. 35
ChromaBASIC	NIBBLE	Vol.3 No. 1	p. 8
LPA MacProlog	MACUSER	Feb. 88	p. 101
Mac Compilers	MACUSER	Feb. 88	p. 267
MacBrain	MACWORLD	Feb. 88	p. 207
MS-BASIC Sound	NIBBLE	Vol.3 No. 1	p. 36
Programmer's Workshop	MACUSER	Jan. 88	p. 203
SemperSoft Modula-2	MACWORLD	Feb. 88	p. 216
Turbo Pascal Tutor	MACWORLD	Feb. 88	p. 216

Databases

4th Dimension	MACUSER	Apr. 88	p. 119
dBASE Mac	MACUSER	Feb. 88	p. 123
dBASE Mac	MACWORLD	Mar. 88	p. 170
Double Helix II	MACUSER	Apr. 88	p. 108
Inside Out	MACWORLD	Feb. 88	p. 219
Insights on FileMaker Plus	MACWORLD	Feb. 88	p. 275
Reflex Plus	MACWORLD	Mar. 88	p. 173
Reflex Plus	MACUSER	Feb. 88	p. 137

Desktop Publishing

Best Ways to Use PostScript	MACUSER	Jan. 88	p. 189
Color Cutter	MACWORLD	Feb. 88	p. 207
Insights on Adobe Illustrator	MACWORLD	Jan. 88	p. 219
Just Write	MACWORLD	Feb. 88	p. 155
LearnWord 3.0	MACUSER	Apr. 88	p. 88
MacTeX	MACUSER	Apr. 88	p. 167
Microsoft Word 3.01	MACUSER	Mar. 88	p. 140
MindWrite	MACUSER	Feb. 88	p. 161
Personal Writer	MACUSER	Mar. 88	p. 206
QUED/M (text editing)	MACUSER	Mar. 88	p. 186
QUED/M (text editing)	MACWORLD	Jan. 88	p. 168
Ready Set Go 4.0	MACUSER	Feb. 88	p. 195
Scoop	MACUSER	Feb. 88	p. 178
Scoop	MACWORLD	Mar. 88	p. 168
Starting with Desktop Publishing	MACWORLD	Feb. 88	p. 253
TeXtures	MACUSER	Apr. 88	p. 167
TeXtures: Typesetting	MACWORLD	Feb. 88	p. 200
The PostScript Difference	MACWORLD	Jan. 88	p. 138
Word Finder (thesaurus DA)	MACWORLD	Feb. 88	p. 206

Engineering

MCS/pal 1.95	MACWORLD	Mar. 88	p. 180

Games

Beyond Zork	MACUSER	Apr. 88	p. 98
Computer Scrabble	MACWORLD	Mar. 88	p. 183
Crystal Quest	MACUSER	Apr. 88	p. 198
Go for the Macintosh	MACWORLD	Jan. 88	p. 158
MSFL Pro League Football	MACUSER	Apr. 88	p. 208
Plundered Hearts	MACUSER	Jan. 88	p. 88
Quarterstaff	MACWORLD	Feb. 88	p. 205
Quarterstaff	MACUSER	Mar. 88	p. 90
Strategic Conquest Plus	MACUSER	Feb. 88	p. 240
Tactical Jet Fighter	MACWORLD	Jan. 88	p. 169
The Fool's Errand	MACWORLD	Feb. 88	p. 210
Trust and Betrayal	MACUSER	Mar. 88	p. 175

General

Design (flowcharts)	MACUSER	Apr. 88	p. 146
Designing Forms	MACWORLD	Feb. 88	p. 291
Expert Systems	MACUSER	Jan. 88	p. 136
Family Roots	MACWORLD	Feb. 88	p. 213
FlowMaster (flowcharts)	MACUSER	Jan. 88	p. 106
Getting Help with Problems	MACWORLD	Jan. 88	p. 121
Handwriting Analyst	MACUSER	Apr. 88	p. 101
Increasing Macintosh Speeds	MACUSER	Mar. 88	p. 109
Instant Expert	MACUSER	Jan. 88	p. 138
Mac-ESP 1.4	MACWORLD	Jan. 88	p. 165
MacSmarts	MACUSER	Jan. 88	p. 146
Macworld Expo Jan 88	MACUSER	Apr. 88	p. 135
MultiFinder Ramblings	MACUSER	Jan. 88	p. 73
ResumeWriter	MACWORLD	Jan. 88	p. 160
SubliMac 2.0	MACWORLD	Jan. 88	p. 165
System Update 5.0	MACUSER	Mar. 88	p. 127
The New Mac System Folder	MACWORLD	Jan. 88	p. 195
Upgrading Your Macintosh	MACWORLD	Feb. 88	p. 137

Graphics

Canvas	MACUSER	Jan. 88	p. 154
Color Paint Programs	MACWORLD	Feb. 88	p. 149
Colorizer	MACUSER	Jan. 88	p. 92
Crystal Paint	MACUSER	Feb. 88	p. 90
Cutting and Pasting Graphics	MACWORLD	Feb. 88	p. 194
DeskPaint	MACWORLD	Mar. 88	p. 174
DeskPaint	MACUSER	Mar. 88	p. 84
Dimensions 1.18 (3-D)	MACWORLD	Jan. 88	p. 153
Draw It Again Sam	MACUSER	Feb. 88	p. 96
Gray Scales	MACUSER	Apr. 88	p. 177
LaserPaint	MACUSER	Jan. 88	p. 168
Mac Charting Tools	MACWORLD	Jan. 88	p. 227
MacCalligraphy	MACUSER	Feb. 88	p. 213
Modern Artist	MACWORLD	Jan. 88	p. 161
PowerPoint (presentations)	NIBBLE	Vol.3 No. 1	p. 12
Read-It! (OCR software)	MACUSER	Apr. 88	p. 216
Read-It! (OCR software)	MACUSER	Feb. 88	p. 90
SuperChroma	MACWORLD	Jan. 88	p. 161

Hardware

Apple PC 5.25 Drive	MACWORLD	Feb. 88	p. 204
AST TurboScan	MACWORLD	Jan. 88	p. 167
DaynaFile (DOS drive)	MACUSER	Jan. 88	p. 94
DaynaFile (DOS drive)	MACWORLD	Feb. 88	p. 204
Dynamic EL (portable Macintosh)	MACWORLD	Feb. 88	p. 211
GCC Personal LaserPrinter	MACUSER	Apr. 88	p. 155
GCC Personal LaserPrinter	MACWORLD	Mar. 88	p. 181
GPIB-SE (instrument control)	MACWORLD	Jan. 88	p. 166
Gray-Scale Scanners	MACWORLD	Jan. 88	p. 111
HP ColorPro Plotter	MACUSER	Feb. 88	p. 98
ImageMaker (35mm slide maker)	MACUSER	Feb. 88	p. 149
Is/One Penworks (input device)	MACUSER	Jan. 88	p. 84
Jasmine Direct Drive	MACUSER	Mar. 88	p. 84
LaserWriter II	MACWORLD	Mar. 88	p. 126
Mac II Memory Expansion	MACWORLD	Mar. 88	p. 136
Mac-101 (keyboard)	MACUSER	Apr. 88	p. 92
MacLarger	MACUSER	Apr. 88	p. 82
Macnifier (OH projector)	MACWORLD	Mar. 88	p. 185
MacTable	MACUSER	Mar. 88	p. 86
Nutmeg/Xerox Page Display	MACWORLD	Mar. 88	p. 176
ScripTEN Laser Printer	MACWORLD	Feb. 88	p. 202
Varityper VT600 (laser)	MACWORLD	Jan. 88	p. 152
Workstations: Mac II vs. Apollo	MACWORLD	Feb. 88	p. 170
Workstations: Mac II vs. Sun	MACWORLD	Feb. 88	p. 165

HyperCard

HyperCard Fields	NIBBLE	Vol.3 No. 1	p. 20
HyperTutor	MACUSER	Apr. 88	p. 90
Searching and Merging Scripts	NIBBLE	Vol. 3 No. 2	p. 32
Switching Data to HyperCard	MACWORLD	Jan. 88	p. 209

Mathematics

Eureka: The Solver	MACUSER	Mar. 88	p. 87

Music

Performer	MACUSER	Jan. 88	p. 94
Professional Composer	MACUSER	Jan. 88	p. 103

Networks

ComServe	MACUSER	Feb. 88	p. 226
Getting Started with MACVAX	MACWORLD	Mar. 88	p. 213
Insights into TOPS Network	MACWORLD	Mar. 88	p. 223
Integrating Mac & VAX	MACWORLD	Mar. 88	p. 140
NetModem V1200	MACUSER	Feb. 88	p. 226
The Ethernet Solution	MACWORLD	Jan. 88	p. 128
Using Networks	MACWORLD	Jan. 88	p. 104

Science

Physics	MACWORLD	Jan. 88	p. 159
Sensei Physics	MACUSER	Jan. 88	p. 90

Spreadsheets

(Excel) Car Cost Calculator	NIBBLE	Vol.3 No. 1	p. 33
(Excel) Database Data Entry	NIBBLE	Vol.3 No. 1	p. 26
(Excel) Macros	MACWORLD	Feb. 88	p. 265
(Excel) Toggle Macros	NIBBLE	Vol. 3 No. 2	p. 30
Selecting a Spreadsheet	MACWORLD	Feb. 88	p. 177

Stackware

Business Class	MACUSER	Mar. 88	p. 166
Flash Cards (HyperCard stacks)	MACUSER	Feb. 88	p. 276
Flash Cards (HyperCard stacks)	MACUSER	Jan. 88	p. 210
Flash Cards (HyperCard stacks)	MACUSER	Mar. 88	p. 220
Flash Cards (HyperCard stacks)	MACUSER	Apr. 88	p. 224
Focal Point	MACUSER	Mar. 88	p. 166
HyperCard Stacks	MACWORLD	Feb. 88	p. 209
HyperGeometer (geometry)	NIBBLE	Vol. 3 No. 2	p. 24
Stacks of Stacks	NIBBLE	Vol. 3 No. 2	p. 8

Telecommunications

Modems (2400 baud)	MACWORLD	Mar. 88	p. 154
The CIS Navigator	MACWORLD	Feb. 88	p. 212

Utilities

AutoSave DA	MACUSER	Jan. 88	p. 110
ComServe	MACWORLD	Mar. 88	p. 177
Data Savers: Backup Programs	MACWORLD	Feb. 88	p. 189
Desktop Express	MACUSER	Jan. 88	p. 180
Disk Tools Plus	MACUSER	Mar. 88	p. 96
DiskFit 1.3	MACWORLD	Mar. 88	p. 175
Easy Access	MACUSER	Feb. 88	p. 256
FastBack	MACUSER	Apr. 88	p. 92
FastBack 1.01	MACWORLD	Mar. 88	p. 175
HFS Backup 2.02	MACWORLD	Mar. 88	p. 175
Icon-it	MACWORLD	Feb. 88	p. 214
Icon-it	MACUSER	Feb. 88	p. 99
MacIrma (IBM 3278 emulation)	MACWORLD	Mar. 88	p. 178
MacTree	MACUSER	Apr. 88	p. 96
QuickDex (DA)	MACWORLD	Feb. 88	p. 215
QuicKeys	MACUSER	Jan. 88	p. 114
QuicKeys 1.0	MACWORLD	Jan. 88	p. 155
Rendezvous 1.3	MACWORLD	Mar. 88	p. 184
Sentinel (data security)	MACUSER	Apr. 88	p. 104
Sentinel (password protection)	MACWORLD	Feb. 88	p. 218
SoundWave	MACUSER	Apr. 88	p. 187
Stepping Out	MACUSER	Mar. 88	p. 195
Stuffit 1.10	MACWORLD	Mar. 88	p. 177
TMON 2.8 (debugger)	MACWORLD	Feb. 88	p. 203

Macintosh Books and Publications

Designing Cards and Drivers for the Macintosh II and Macintosh SE

Inside Macintosh Volume 5
This publication covers the programming tools available for the Macintosh II and the Macintosh SE, color, sound, and the Apple Desktop Bus™ protocol.

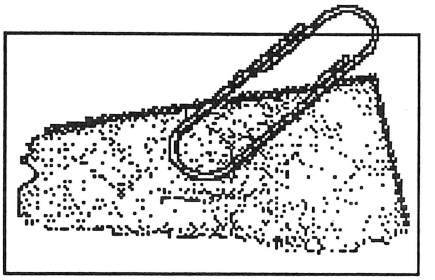
Inside Macintosh X-Ref
This is an index to all five *Inside Macintosh* volumes.

Technical Introduction to the Macintosh Family
This book provides an overview of the hardware and software architecture of all Macintosh computers. It compares the classic Macintosh design with the open-architecture design of the Macintosh SE and the Macintosh II, and addresses methods of maintaining compatibility among the various machines. In addition, the book explains the design and operation of all the Macintosh computers, describes the user interface in detail, and outlines the A/UX operating system.

A Programmer's Introduction to the Macintosh Family
ImageWriter LQ Reference
These two books will be published later this year.

For more information, contact:
Addison-Wesley Publishing Company
Route 128
Reading, MA 01867
(617) 944-3700

Project Reports



This section of *Wheels for the Mind* presents many of the projects that are being developed at various colleges and universities. The listings are not restricted to Consortium schools or to schools in this country; you will find projects that are being done at universities all over the world.

Some of the items listed here are just in the beginning stages, while others are completed and ready for distribution.

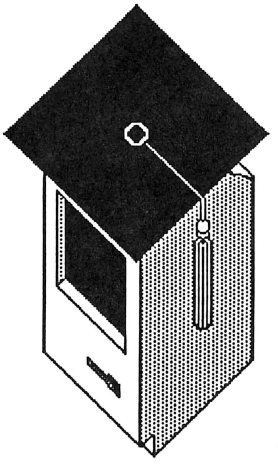
Rather than accumulating more and more reports, repeating them issue after issue, we are trying to list only new projects, or projects that have recently been further developed or updated.

Wherever possible, we have tried to list an appropriate contact, in case you are interested in finding out more about a listing.

One of the original intents of organizing educational institutions around the Macintosh was to provide opportunities for creative people to share their expertise and talents. We are sure that you will find much of that creativity in evidence here.

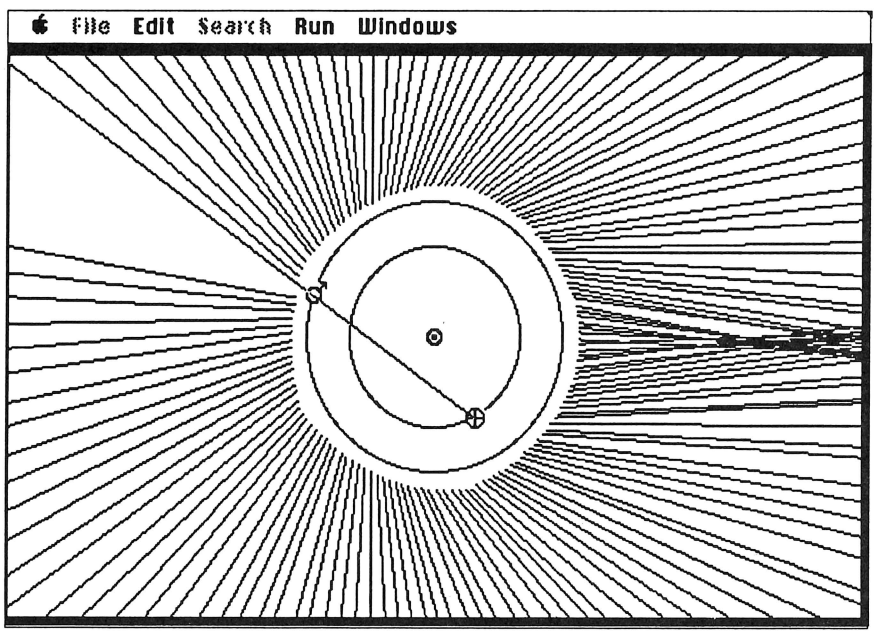
Please feel free to collaborate on developing an application, to develop one on your own, or simply to use one that was developed by one of your colleagues.

Most important, let us know what YOU are working on. Perhaps there are others around the world who might like to work with you! Send us a description of your project.



Astronomy Concepts

These are simple graphical programs intended to demonstrate concepts that some students find difficult. Programs completed to date are Ellipse, Kepler 3, and Retrograde. Ellipse is interactive—the student enters various eccentricities. Kepler 3 is a demonstration of Mercury, Venus, Earth, and Mars in orbit. Retrograde (see figure below) provides an animation of the Earth passing Mars, illustrating changing lines of sight. The programs are written in Microsoft BASIC, and are compiled and linked (no license is required for distribution to students). Copies are available (send disk). Source code is also available on a cooperative basis.
Current status: Three programs completed.
Contact: Dr. Lawrence P. Staunton, Department of Physics and Astronomy, Drake University, Des Moines, IA 50311; (515) 271-3033



Biology

An Auto-Tutorial in Recombinant DNA Technology

This "auto-tutorial" addresses a wide audience, from students in introductory biology courses to professionals wishing to expand their scientific vocabulary. The tutorial is composed of three major sections that emphasize key aspects of genetic engineering: 1) the methodology of cloning DNA; 2) examples of recombinant DNA products currently being developed and/or marketed; and 3) societal implications that arise from the potential of recombinant DNA. In addition, the tutorial has a review section about basic biological processes such as transcription and translation. The tutorial contains its own Help section, Index, and Bibliography. It uses the powerful tools of HyperCard to link concepts between the sections. Browsers can click to obtain definitions, cross-references, or supplemental information—all without losing their place in the tutorial. Animation, many diagrams, cartoons, and self-tests are used throughout the tutorial to aid the reader in fully comprehending the material. System requirements: Macintosh Plus with two 800K drives and HyperCard software.
Current status: Available through site licensing or by purchasing individual copies ($20.00 for the single 800K disk).
Contact: Jon C. Glase, Senior Lecturer, Section of Neurobiology and Behavior, Cornell University, 1130 Comstock Hall, Ithaca, NY 14853; (607) 255-1304

Chemistry

Alchemist

Alchemist, a program to assist students studying organic chemistry, is currently under development by Joyce C. Brockwell, John H. Werner, and Steven J. Townsend of the Department of Chemistry at Northwestern University. The program is a chemistry "calculator" that helps students solve many of the more difficult problems in elementary organic chemistry. Students are able to enter any molecule and then ask the program various questions about it. This ability makes the program a major departure from the standard drill and tutorial offerings currently available. Development of Alchemist was assisted by grants from the Alfred P. Sloan Foundation, IBM, and Apple Computer.

To communicate with the program, students use a MacDraw-like palette to draw molecules on the screen. They can draw simple atoms and bonds, as well as features such as rings, heavy isotopes, charges, and nonbonding electrons. Once the molecules are drawn, the student uses Alchemist's menus to ask various questions about them. The information currently available includes resonance structures, IUPAC names, NMR spectra, Newman projections, Lewis dot diagrams, and hydrogen pKa's.

The program is also able to predict the products of reactions involving molecules drawn on the screen or entered from a large reagent list. The products are predicted by mechanistic reasoning, which allows the student to examine the reaction mechanisms, as well as the products. Among the reactions Alchemist currently knows are S_N1, S_N2, E1, E2, organometallics, carbonyls, acid-base reactions, bromonium formation, and oxidative cleavage.
The production of this organic chemistry study aid has pointed out the power inherent in a program that could actually perform the role of a tutor. Such a "machine tutor" would be capable of working through and explaining many of the problems presented in an elementary textbook; it could answer questions, spot errors, check student work, and provide clues and explanations; all in a style that avoids the tedium of prerecorded responses. Means of incorporating these features are being investigated as development of the program proceeds. System requirements: Macintosh with at least 512K of memory. Alchemist performs best on a Macintosh Plus or Macintosh SE. It also runs, but is slightly buggier, on a Macintosh II.
Current status: Ongoing. A prerelease version of the program and manual is available from the developers for $25.
Contact: J. C. Brockwell, Department of Chemistry, Northwestern University, Evanston, IL 60208; (312) 491-3440

Periodic Table Tool

The Periodic Table Tool (PTT) is a problem-solving tool that features the periodic table and opens the fundamental organization of the chemical and physical properties of the elements for electronic exploration. The PTT allows the creation and manipulation of chemical databases and can display elemental properties in either graphical or tabular form. This makes it suitable for use by an instructor at either the secondary or collegiate level, as a basis for discussion within the classroom or as a means for providing "discovery" learning for students outside of class time.

With the flexibility to design and modify several periodic tables simply by dragging the elements to their desired position and entering their individual properties, the PTT lets users create comparative table models consisting of any number of elements, which may be accessed or changed as needed. Properties such as melting point, electronegativity, or density may be added or deleted, and individual data for up to 125 elements is easily edited. Boolean searches for elements that satisfy user-defined criteria for chemical and physical properties may be used to discover relationships between elements and their properties, and this data may be used as the basis for new tables. User-defined equations may also be used to analyze data. Numerical data may be plotted on an X-Y graph for the entire table, or a plot may restrict data to traditional groupings—such as the representative elements, transition metals, or halogens—and up to ten graphs may be saved within each database for immediate reference and display.

Current status: Completed.
Contact: James F. Johnson, Assistant to Chairman, Department of Chemistry, 247 Nieuwland Science Hall, University of Notre Dame, Notre Dame, IN 46556; (219) 239-5659 or Mark B. Johnson, User Services, Computing Center, University of Notre Dame, Notre Dame, IN 46556-0539; (219) 239-5600; BITNET: <CDTAXW@IRISHMVS>; AppleLink: A72

Chemistry

Huckel Molecular Orbitals

The program Huckel Molecular Orbital (HMO) calculates energy eigenvalues, eigenvectors (molecular orbital coefficients), pi-electron change densities, and pi-electron bond orders for conjugated and aromatic molecules, using the assumptions of simple HMO theory. A window is used to enter the resonance structure of the molecule. Along the left side of the resonance structure window is a MacDraw-like palette used to select the type of atom or bond to be entered. The results can be viewed in a results window and printed. The application can be used by anyone who can draw the resonance structure of a molecule. System requirements: Macintosh computer with at least 512K of RAM.
Current status: Version 2.1 complete.
Contact: John J. Farrell, Chemistry Department, Franklin and Marshall College, P.O. Box 3003, Lancaster, PA 17604-3003; (717) 291-3803

Computer Science

Quick Procedure

Quick Procedure is an FKEY that writes out a blank procedure for a specified procedure call. For example, the FKEY will take the procedure call "SortList(list, elements, key);" and write out the blank procedure:

```
"procedure SortList(list:TYPE;
            elements:TYPE;
            key:TYPE);
begin
 end;"
```

This saves the user typing time, and ensures that the parameters are specified in the correct order in the blank procedure definition. The FKEY currently works only with the Lightspeed Pascal editor, but will soon work with any editor.
Current status: Version 0.9 completed.
Contact: Bryce Jasmer, Oregon State University, 329 Finley, Corvallis, OR 97331

VLSI Magician

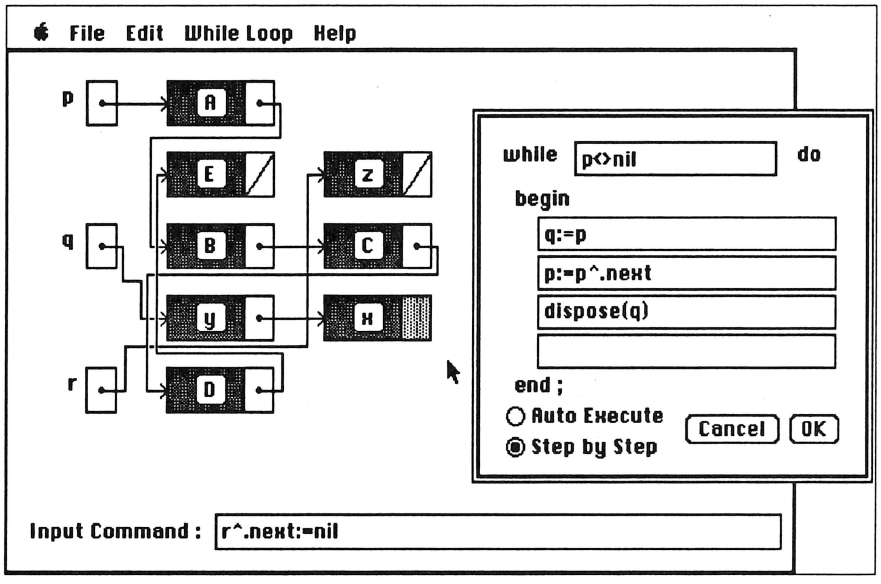
VLSI Magician is used to prepare custom circuit layouts. Three design technologies are supported: NMOS, CMOS, and a single-metal-layer thick film hybrid circuit technology. The user designs the circuit graphically on the screen, and the application translates these graphs into a text file. This file is suitable for input to the Magic layout editor from the University of California at Berkeley for design rule checking and circuit extraction. VLSI Magician has its greatest utility in the design of logic primitives, small groups of primitives, and hybrid circuits. A computer lab with several Macintosh computers can make a significant user-load reduction on expensive color CAD workstations. This software has been applied at Drexel in a microelectronics course sequence for the layout of small CMOS circuits, and in a thick film hybrid circuit fabrication course. System requirements: Macintosh 512K Enhanced, Macintosh Plus, Macintosh SE, or Macintosh II.

Current status: VLSI Magician is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Kevin J. Scoles, Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104

Pascal Pointers

Pascal Pointers is intended to provide users with a mechanism for learning to use pointers and dynamic variables in Pascal. The concept arose from the great deal of difficulty students have exhibited in learning this material in programming courses. This material is fundamental for further computer science work, yet often is not learned by students, partially because it is easier for the programmer to find an alternative way to implement a program than it is to learn the more elegant (and often more efficient) methods associated with dynamic variables.

The package is designed as follows: The user is supplied with a display that shows an initial data structure. The display is similar to that used in most Pascal textbooks. The user is allowed to enter commands from a subset of Pascal commands that use pointer variables. As each command is entered, the display is updated to demonstrate the effect of the command on the given structure. Syntax and semantic errors are detected by the interpreter, with appropriate diagnostics. Some on-line help is also provided. A set of lessons is available, each of which provides some initial structure, and asks the user to alter it in some way. The user is not required to perform the lessons, however; this package is designed for use as a general tool that will correctly interpret and display any sequence of pointer-type commands.



Pascal Pointers may be used in the classroom for demonstration. Further, the instructor may create lessons for student use, composed of an initial configuration and a brief set of instructions. This generality allows the package to be used as a teaching tool, and tailored to instructors' individual preferences.

This package has proved useful to students wishing to learn about or to refresh their knowledge of the use of dynamic variables and pointers. It provides a visual conceptual model similar to that found in a typical Pascal or data structures text, yet allows the interaction and "What if?" aspects that a textbook lacks. It encourages learning by experimentation, because the effects of each command are clearly displayed. This differs from the usual programming environment, which can crash, loop infinitely, or produce undesired results without allowing the programmer to know why.

This package is also useful as a reference tool for the programmer trying to write or debug a section of code using dynamic variables.

Current status: Pascal Pointers is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Engineering

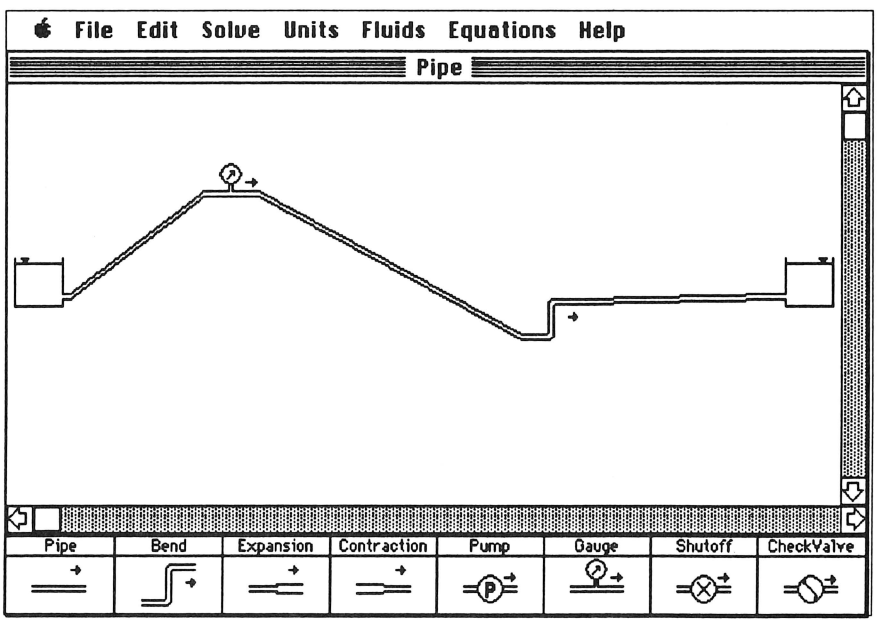
Fluid Mechanics

Three instructional applications for Fluid Mechanics have been developed: PotFlo (Potential Flow) allows construction of simple solutions of the Laplace equation by superposition of elementary potential flows of variable strength; CompFlo (Compressible Flow) replaces a complete set of tables and shock charts for the solution of problems in the steady one-dimensional flow of perfect gases with arbitrary specific heat ratio (The results obtained are useful in the study of isentropic and adiabatic flows, involving normal and oblique shocks); and Nozzle (1D Laval nozzle flow) is a simulation of a Laval nozzle operating with area ratio and pressure ratio adjustable by the operator. A fourth application—AirFoil (Wave lift and drag)—is about half completed.
Current status: Three applications completed and in use. Fourth under development.
Contact: Prof. Abraham S. Berman, Aerospace Engineering and Mechanics, 119 B Akerman Hall, University of Minnesota, Minneapolis, MN 55455

Pipe Flow

Pipe Flow facilitates the design of a pipeline carrying the steady-state flow of an incompressible liquid between two reservoirs. The program is designed to be used both as a training aid by engineering students and as a working tool by practicing engineers.

Engineering courses for which the Pipe Flow program would be a valuable supplement include fluid mechanics, hydraulics, pumping systems, and mass transport. Practicing engineers could utilize the



Engineering

Pipe Flow program for the steady-state component design of small water supply systems, refined petroleum product pipelines with only a single pump station, and single chemical pipelines.

Pipe Flow uses a paradigm of graphic objects that can be easily manipulated with the standard Macintosh mouse techniques of pointing, clicking, and dragging. This intuitive operating style permits the user to quickly and easily contruct on the screen a detailed pipeline between two reservoirs.

Pipeline appurtenances such as valves, bends, pressure gauges, a contraction, an expansion, and a pump or turbine can be included as an integral part of the pipeline. Actual pipes can consist of numerous standard construction materials of various diameters. Flow resistance coefficients can be specified by the user or selected from a list of standards built into the program. The analysis performed by the Pipe Flow program is limited to the steady-state, isothermal flow of an incompressible liquid through a single pipeline joining two reservoirs. The pipeline may contain at most one contraction, one expansion, and one centrifugal pump or turbine. If both a contraction and an expansion are used, the pipeline may contain up to three different pipe diameters. The total number of pipeline appurtenances (bends, valves, pressure gauges, expansion, contraction, pump) is limited to about 20.

Unknowns are limited to pressure head, elevation head, flow rate, or diameter. Only one unknown can be specified at a time. Pressure gauges require the elevation of the pipeline at the gauge to calculate the pressure head.
Current status: Pipe Flow is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

History

Salem Data Set

This data set, which includes tax records, petitions, Salem Village committee lists, and names and residences of accusers and witches, is designed for use with Data Desk, a statistical and graphics software program. It can be incorporated into courses dealing with the Salem witchcraft trials, or into survey American history, American colonial history, or historical study methodology courses. The Salem Data Set is an excellent way of adding a microcomputer component to the history curriculum and of dem-

onstrating to undergraduates and graduate students the techniques of quantitative history. A manual describes use of the data set in a classroom and contains illustrations of data output.
Current status: Completed. Available from Tulane Computing Services, Faculty Computing Resource Library.
Contact: Professor Richard B. Latner, Department of History, Newcomb College, Tulane University, New Orleans, LA 70118; (504) 865-5588; BITNET: CBOX @ TCSVM

Management

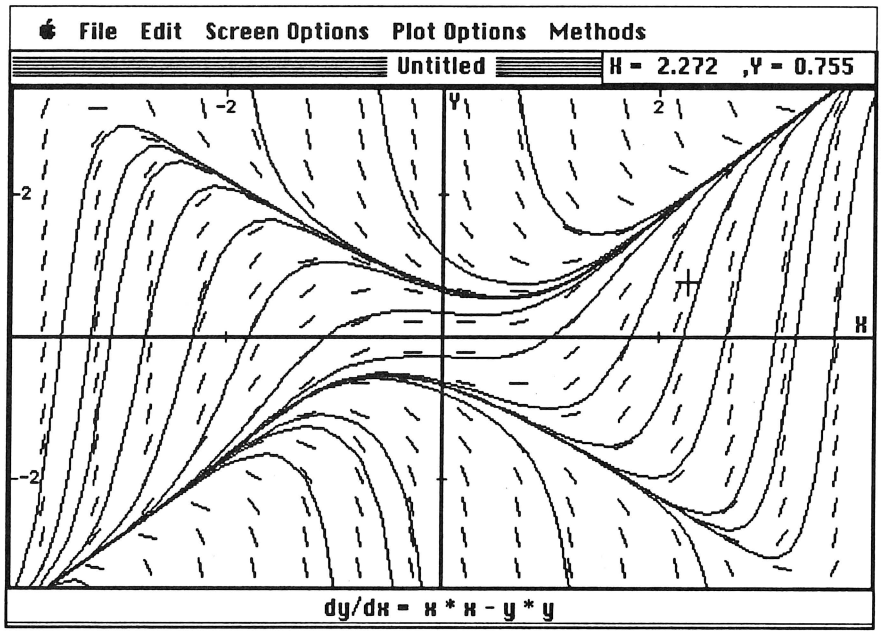
The Production Planning and Control Game

ABC Inc. is an educationally oriented manufacturing planning and control simulation. Interacting with the program, a student will plan and execute the production and purchasing of all parts needed by a hypothetical manufacturing company. The system will include integrated Material Requirements Planning and Shop Floor Control Windows. Students will use the program to do production planning, master scheduling, materials planning, short-range capacity planning, and shop floor execution. An accompanying manual containing the description of the ABC Inc. with various computer exercises is also planned.
Current status: Expected completion in January 1989.
Contact: James Ward, Associate Professor, School of Management, Purdue University, Krannert Building, West Lafayette, IN 47907; (317) 494-4489

Mathematics

Exploratory Graphics for Scalar Differential Equations

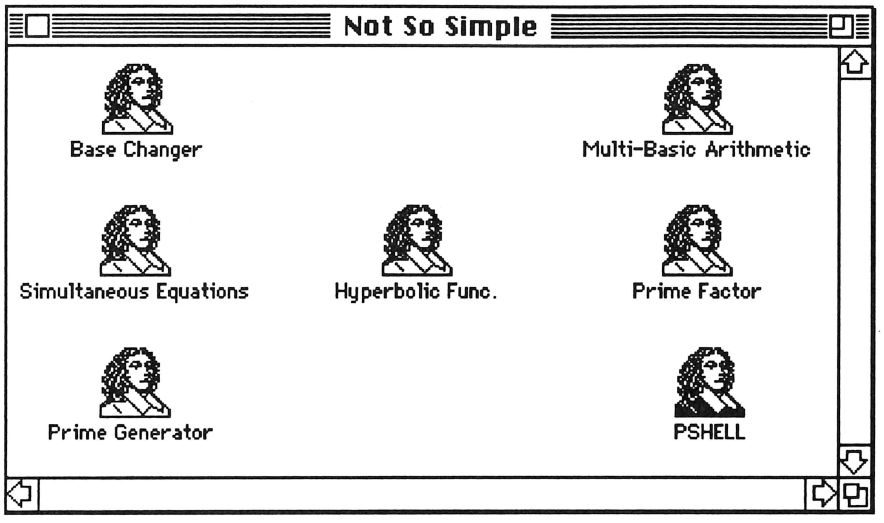
Differential Equations is an exploratory graphics tool for the study of scalar first-order differential equations of the form $dy/dx = f(x,y)$. It is designed for the teacher in the classroom and the student outside of the classroom. The user has control over the viewing rectangle, direction field display, various plotting options, and which of several classical approximation methods to use for plotting. The function $f(x,y)$ is specified by typing the desired expression, using a standard syntax for expressions (for instance $f(x,y) = x^2 - y^2$ is entered as 'x*x - y*y' or 'x≠2 - y≠2' at the keyboard). On-line help is available, and the ability to save the application at any time enables teachers to prepare several screens and thereby save valuable class time. System requirements: With modest demands, the application will run on a Macintosh 128K;



more serious work requires at least 512K of memory.
Current status: Differential Equations is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Professor Herman Gollwitzer, Department of Mathematics and Computer Science, Drexel University, Philadelphia, PA 19104

Not So Simple Arithmetic

Not So Simple Arithmetic consists of six parts: Base Changer, Multi-Basic Arithmetic, Simultaneous Equation Solver, Hyperbolic Function Table, Prime Factors, and Prime Generator. Base Changer permits the entry of a number whose radix (base) lies in the range 2 to 16 (inclusive) and the obtaining of an equivalent number of different radix within the same range. Multi-Basic Arithmetic provides for the addition, subtraction, multiplication, or division of numbers with radices in the range 2 to 16 inclusive, with the exception of base 10. Simultaneous Equation Solver solves linear simultaneous equations involving two, three, or four unknowns. Any mistakes will be explained to the user and, where appropriate, suggestions made. Hyperbolic Function provides a means of obtaining the hyperbolic functions or inverse functions of the values introduced as inputs. Prime Factors provides the means of obtaining the prime factors of the numbers entered. The Prime Generator program generates a series of prime numbers beginning with a selected number.



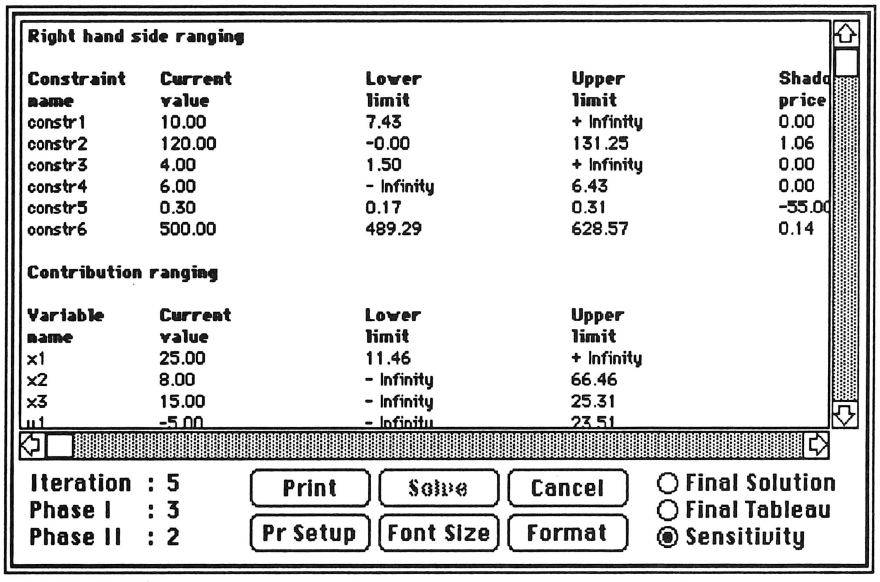
Current status: Not So Simple Arithmetic is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Big ALPAL

Big ALPAL was developed to solve linear programming problems on the Macintosh. The program is an extension of the ALPAL program, which was designed for the Macintosh 128K. The size of the model that can be solved depends on the memory available, which is sensed by the program. The number of decision variables and the number of structural constraints possible are each in excess of 200 on a Macintosh 512K and more than 300 on a Macintosh Plus.

The software supplies the necessary slack, surplus, and artificial variables to analyze a linear programming model, as well as the nonnegativity constraints. Sensitivity analysis for the objective function values and the right-hand sides is available. A menu furnishes shortcuts for entering variable names, constraint names, and right-hand sides in large problems. Various outputs are available, including the simplex tableaux used, the optimal solution, and the sensitivity analysis.
Current status: Big ALPAL is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Jon Burton, Department of Management, Drexel University, Philadelphia, PA 19104

Mathematics



Right hand side ranging

Constraint name	Current value	Lower limit	Upper limit	Shado price
constr1	10.00	7.43	+ Infinity	0.00
constr2	120.00	-0.00	131.25	1.06
constr3	4.00	1.50	+ Infinity	0.00
constr4	6.00	- Infinity	6.43	0.00
constr5	0.30	0.17	0.31	-55.0
constr6	500.00	489.29	628.57	0.14

Contribution ranging

Variable name	Current value	Lower limit	Upper limit
x1	25.00	11.46	+ Infinity
x2	8.00	- Infinity	66.46
x3	15.00	- Infinity	25.31
u1	-5.00	- Infinity	23.51

Iteration : 5
Phase I : 3
Phase II : 2

Print | Solve | Cancel | Pr Setup | Font Size | Format

○ Final Solution
○ Final Tableau
◉ Sensitivity

FEMG—A Finite-Element Mesh Generator

FEMG is an interactive application for generating and editing two-dimensional finite-element meshes. The elliptic grid generation used requires only that the boundary nodes be specified, which is done with the mouse. FEMG supports triangular and quadrilateral elements of degree one and two. FEMG is also an editing tool that allows for moving nodes and for local element refinement. It can generate the incidence list and, upon specification of the coordinate system, the nodal locations.

The engineer and applied scientist who deal with medium (300 elements) and large (3,000 elements) finite-element meshes, will find this tool most useful. It can also help teach the art of making finite-element meshes.

FEMG was used to generate a large finite-element grid for the English Channel and the southern part of the North Sea. Only a few hours were needed to generate the grid, though it has 990 nodes and 1,762 elements. Most of the time was spent digitizing a map of the area to serve as a guide for the grid generation. The mesh was later used for the simulation of the tide in that area. System requirements: Macintosh with at least 512K of RAM. Current status: Completed and available through Kinko's Academic Courseware Exchange (800-235-6919).

Medical

A PharMACopia of Drug Actions

The project consists of two parts. The first is a relational database for drugs. The user can access several types of information about a particular drug, including (1) trade and generic names, (2) proposed mechanism of action, (3a) disposition, (3b) sites of action, (3c) adverse effects, (4) potential drug interactions, and (5) clinical uses. Implementation of this as a HyperCard stack is complete, and a fully working model with 50 drug entries exists. Complete data has been entered for about 20 drugs; information on the remaining 30 drugs will be fully entered by the middle of February. Work is proceeding on a HyperCard stack of pharmacology questions and answers, including detailed explanations for each answer.

Current status: First part complete; second near completion.

Contact: Dr. L. Lichtblau, Department of Pharmacology, 3-260 Millard Hall, 435 Delaware Street SE, University of Minnesota, Minneapolis, MN 55455

Miscellaneous

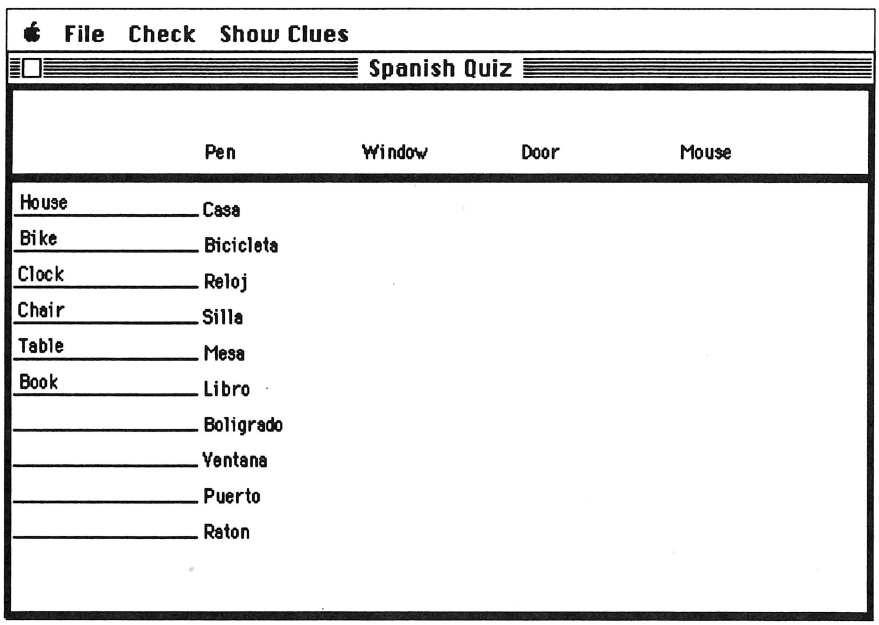
MatchMaker and MatchMaker Utility

MatchMaker interactively presents, checks, and scores matching-type quizzes. To match words to definitions, you physically drag them together on the screen. You can then request that the program find and mark any incorrect pairings. MatchMaker Utility is the program used to create the quizzes that are run by MatchMaker.

The screen is divided into two rectangular areas. The smaller of the two areas, at the top of the screen, contains a selection of up to ten words. In the larger box below, there is a list of definitions, each preceded by a short underlined blank space.

You take the quiz by "filling in the blanks"; that is, by using the mouse to drag the words one at a time from the top of the screen down onto the lines preceding the definitions. A word will automatically "snap" into position at the left end of the line once it has been placed on the desired location.

Answers can be checked in two ways: selecting Check My Work from the Check menu will tell you how many right, wrong, and unanswered questions there are; while the Quick Check menu item simply places a checkmark next to each wrong answer.



MatchMaker Utility enables you to create quizzes. The program opens a window containing three rectangles, or edit fields. The window represents a question "blank," and you type a word, a definition, and a hint/clue for each question into the edit fields. Pressing Return will advance you to the next field. A pair of buttons at the bottom of the screen advance you to the next question, or take you back to the previous one. MatchMaker displays hints in a scrollable text box, so you can make hints of any length. If you wish, you may draw attention to certain words in the hint section by making them appear in boldface. MatchMaker Utility also allows the option of scrambling the words, definitions, or clues.

Current status: MatchMaker and MatchMaker Utility are available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services Drexel University, Philadelphia, PA 19104

Statistical Mapping Program

The main aspect of the project is the ability to display data on scanned geographic maps. Statistics such as city populations and latitudes and longitudes can be read from a spreadsheet file and placed by the computer on the map (MacPaint format) of the region of interest. The program provides an easy way to translate latitudes and longitudes into pixel coordinates. It also has the ability to deal with boundary delimited maps (regions).

Current status: In development; some portions are available for testing.
Contact: Paulo Ca, Visiting Assistant Professor, Operations Research, Cornell University, School of OR & IE, Upson Hall, Ithaca, NY 14853; (607) 255-5360

Symbiosis

Symbiosis is an application that allows one to explore and experiment with an iterative Prisoner's Dilemma game. The goal is to design the strategy that earns the most points in the long run while interacting with a variety of other strategies designed by other students or the instructor. One or more "entities" (representing the strategies) interact repeatedly during a game, at each point choosing whether or not to "cooperate" with each other. Points are earned depending on what each entity chose as a response following a matrix based on a Prisoner's Dilemma situation. Users may also be personally involved; the program will prompt them for responses when required. The entities interact on a grid of 1,000 cells, and may "reproduce" if they are successful at earning points, or "die" if they are not. Newborn entities use the strategy of their parents; thus, a given strategy may have one or many entities, and population dynamics can play a role in the results obtained. Spatial dynamics may also be explored, by only allowing entities to interact with their immediate neighbors. Other areas of possible exploration include "noise" in the communication between entities, the effects of "taxing" environments, and evolution of mutating strategies. Up to nine strategies may be in a game at one time. Common strategies such as RANDOM, TIT FOR TAT, or TIT FOR TWO TATS are easy to create, and strategies may be password protected and saved to a disk file. The program keeps track of points that are earned by strategies, and summary graphs may be made and exported via the Clipboard. System requirements: Macintosh 512K Enhanced with one 800K drive or more powerful system.
Current status: In beta testing.
Contact: Professor Jane Gibson, Biochemistry Dept., Wing Hall, Cornell University, Ithaca, NY 14853; (607) 255-2432
or
Ron Beloin, Research Support Specialist, Ecosystems Research Center, BTI, Tower Rd., Cornell University, Ithaca, NY 14853; (607) 257-2030

Miscellaneous

Menu Analysis

"Much of what is taught is neither highly technical nor complicated: simple concepts that are daunting only in their unfamiliarity. It is better to save lecture and class time for more interesting matters." This is the motivation behind Menu Analysis. The resources of Microsoft Excel have been used to create an interactive self-teaching spreadsheet; the student learns at his or her own pace, with as many repetitions as are necessary. The menu analysis is a simple way of systematically looking at a sales mix in terms of profitability and popularity. The model shows the potential for such applications in a wide range of areas, from business to engineering and mathematics.

At present the authors have begun work on building a dynamic template that deals with menu pricing models.
Current status: Completed; contact the authors.
Contact: David Romm and Mark Talbert, Asst. Professor/Teaching Assistant, Hotel Administration, Cornell University, 326 Statler Hall, Ithaca, NY 14853; (607) 255-5292

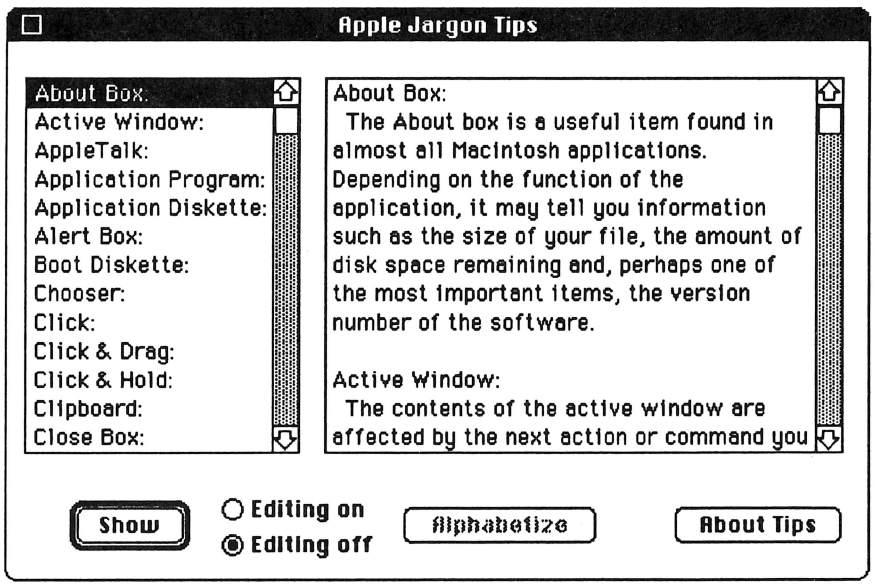
Krannert Executive Education Electronic Link

KEEE-link is a Macintosh-interface front end to any UNIX mainframe for use in sending electronic mail, filing, and conferencing. The electronic mail is compatible with UNIX mail, allowing network and UUCP mail with a Macintosh user interface. Electronic conferencing is compatible with the *notes* Usenet News conferencing software. KEEE-Link provides the same services as AppleLink for the Krannert Graduate School of Business's NCR Tower UNIX system. The UNIX portion of the KEEE-Link software has been ported to other UNIX systems with great success. A port to A/UX is forthcoming, and an MS-DOS front end is under study.
Current status: Completed and ongoing.
Contact: W. Malcolm Duncan, Computer Systems Administrator, Purdue University, Krannert Center for Executive Education, W. Lafayette, IN 47907; (317) 494-7700

Macintosh Shortcuts and Tips

During the last few years, Drexel University's Macintosh consultants have responded to thousands of requests for assistance from both novices and experienced users. Answers to some of the most frequently asked questions have been included in a collection of shortcuts and tips for using the Macintosh and some of its software. Organized into eight different categories—Apple Jargon Tips, FileMaker Tips, Hardware Tips, Nifty Things Tips, Printing Tips, Space & Storage Tips, Spreadsheet Tips, and Term Paper Tips—these helpful hints are now stored in several desk accessories that can be individually installed in the System Folder on a disk using Apple's Font/DA Mover.

After a Tips desk accessory is loaded, it is activated by pulling down the Apple menu and selecting it by name—for example, Apple Jargon Tips. On the left-hand side of the desk accessory, there is a list of terms, and on the right, a set of descriptions. Selecting a term and clicking on the Show button scrolls the corresponding description into view on the right-hand side of the desk accessory.



Miscellaneous

The structure of a Tips Desk Accessory allows users to browse through either the list of tips or the tip descriptions, looking for information. If you are unfamiliar with the information, you may want to scroll through the tip descriptions to get some idea of the kinds of tips already included in the desk accessory. Once you are familiar with the descriptions, you may want to just browse through the list of titles until you find the one you want. You can then select that title and ask the Macintosh to "Show" you the associated description. Sometimes you will discover that simply finding the title of the tip you are looking for will help you to remember the information you want.

The contents of each desk accessory can be edited, so tips can be added. Similarly, if some tips are too elementary, they can be eliminated. Along with the desk accessories containing the short-cuts and tips, there is an empty desk accessory—Tips—that can be used for a variety of purposes (for example, a special-purpose collection of tips, a telephone and address book, a simple outliner, a list organizer, a foreign verb quiz, a general study aid, a set of quick notes for laboratory courses, or a bibliography organizer for writing papers). Documentation is included.

Current status: Tips Desk Accessory is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

ENVIRON

The control of the environment in a library is the single most cost-effective method of prolonging the useful lives of books. With this in mind, graduate student Geoff Yuen and library conservator Frank Farby developed a Microsoft BASIC program to analyze data gathered during a library environmental monitoring project. The program stores temperature and humidity data recorded by hygrothermographs placed within the library. Data is recorded over a 24-hour period, 365 days a year. So far, the monitoring has been carried out over a two-year period.

The software program analyzes daily fluctuations, as well as maximum and minimum levels in temperature and relative humidity, and prints a clear graph of these changes. With the information recorded by this program, it is easy to track variations in the library's environment as a result of seasonal changes. With rapid printout of temperature and relative humidity extremes, it is also possible to keep track of when and how often the temperature and relative humidity in the libraries reach unacceptable levels. And

the program's built-in graphing ability makes it possible to pinpoint any recurring patterns (daily, weekly, or monthly) in the environment that could impact negatively on long-term preservation of library collections. Eventually, the information generated from this computer program will be incorporated into a grant proposal to upgrade and improve environmental controls within the library.

Current status: Ongoing.

Contact: Frank Farby, University Libraries Conservator, Case Western Reserve University Libraries, 11161 East Boulevard, Cleveland, OH 44106; (216) 368-3465

Nutrition

MacDiet—Nutrient Analysis Program

Food intake is analyzed to determine the kilocalorie and nutrient content for three days per file. Foods are selected by name within 16 food groups, including fast foods. Foods and nutrients can be added, deleted, and/or modified by the user. The food and nutrient database supplied with MacDiet covers approximately 1,000 foods and 18 nutrients: kilocalories, protein, carbohydrate, fat, vitamin A, thiamin, riboflavin, niacin, folic acid, vitamins B6 and C, and minerals Fe, Na, K, Ca, P, as well as cholesterol and alcohol.

Personal data (age, sex, height, weight, and frame size) is used to establish nutrient goals. If the user is interested in planning a diet with particular nutrient goals, up to nine nutrients can be selected and monitored during menu planning.

The main menu screen provides the total amounts of each of the nutrients consumed by meal, as well as for the total day's food intake. In addition, reports can be generated that compare the nutrient intake to the Recommended Dietary Allowances and the Dietary Goals.

Nutrition

Energy needs can be calculated two ways: an energy factor that's an estimate of the individual's activity level can be entered; or actual activities for 24 hours can be selected from the more than 70 activities listed in the program. The Activity Report calculates the ideal weight and compares actual to ideal. Estimated weight changes based upon kilocalorie intake and activity level are calculated for a day, a week, and a month, and plotted for a year.

MacDiet is used in our lower-level courses in nutrition and food and in our nutrition course for nonmajors. Students record their food intake and activities, use the program to evaluate their diets, and then write a report describing the quality of their diet and how they could improve it. In the food science courses, MacDiet is used to compare nutrient levels of various food products.

Current status: MacDiet is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Shortie McKinney, Nutrition and Food Sciences, Drexel University, Philadelphia, PA 19104

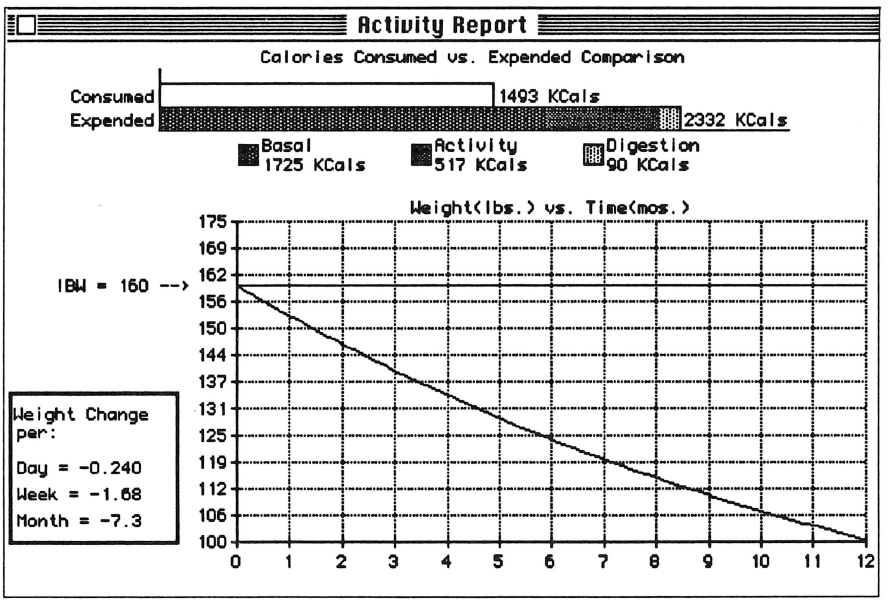
MacDiet Professional

MacDiet Professional is an enhanced and more powerful version of MacDiet. Food intake is analyzed to determine the kilocalorie and nutrient content for seven days per file. Foods are selected by name within 19 food groups, including baby foods, dietetic foods and supplements, fast foods, and ingredients and spices. Foods and nutrients can be added, deleted, and/or modified by the user. The food and nutrient database supplied with MacDiet Professional covers approximately 2,500 foods and 24 nutrients: kilocalories, protein, carbohydrate, fat, vitamin A, thiamin, riboflavin, niacin, folic acid, vitamins B6, B12, C, D, and E, minerals Fe, Zn, Ca, P, as well as polyunsaturated and saturated fat, cholesterol, and alcohol. The impact of adding foods to a diet plan can be evaluated prior to addition. This is helpful in planning restrictive diets.

Personal data (age, sex, height, weight, and frame size) are used to establish nutrient goals. If the user is interested in planning a diet with particular nutrient goals, up to nine nutrients can be selected and monitored during menu planning, and the minimum and maximum goals of each nutrient can be determined by the user.

The main menu screen provides the total amounts of each of the nutrients consumed by meal, as well as the total day's food intake. In addition, the following reports can be generated: Diet Order Comparison, Recommended Dietary Allowances Comparison, Dietary Goals Comparison, Food Group Totals, and Nutrient Contribution by Food Group.

Energy needs can be calculated in the same two ways as with MacDiet.

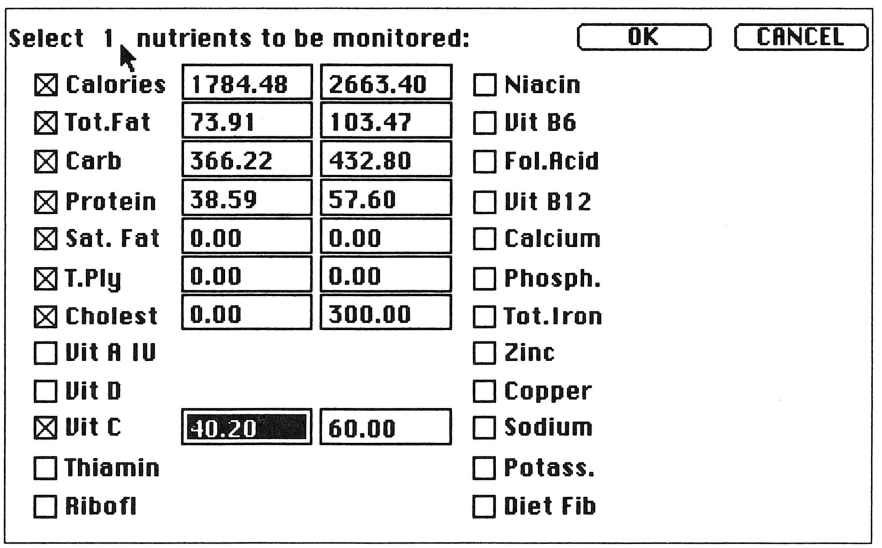


MacDiet Professional is used in our upper-level courses in nutrition and food. Students use the program for diet analysis of diet histories and for diet planning in therapeutic and advanced nutrition courses. MacDiet Professional is also being used to calculate diet records for research studies evaluating a variety of age groups.
Current status: MacDiet Professional is available directly from Drexel University.
Contact: Dr. Shortie McKinney, Nutrition and Food Sciences, Drexel University, Philadelphia, PA 19104

Nutrition

MacDiet Expanded Database

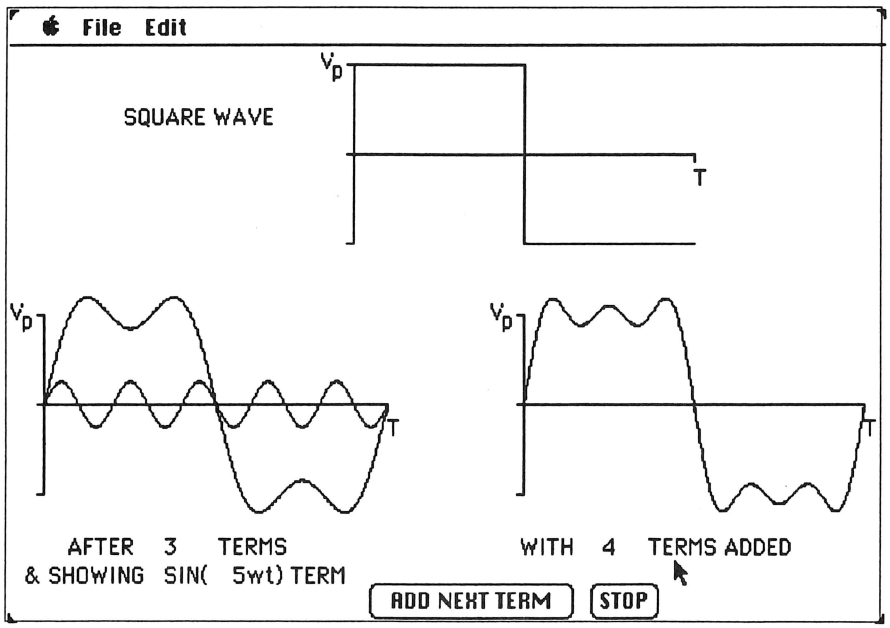
The MacDiet Expanded Database was designed to be used with the MacDiet program. Although the database supplied with MacDiet contains approximately 1,000 foods with 18 nutrients each, the MacDiet Expanded Database contains approximately 2,500 foods, with 24 nutrients per food. Three additional food groups are included (baby foods, ingredients and spices, and nutritional supplements) for a total of 19 food groups. The nutrients included are kilocalories, protein, carbohydrate, fat, vitamin A, thiamin, riboflavin, niacin, folic acid, vitamins B6, B12, C, D, and E, minerals Fe, Zn, Na, K, Ca, and P, as well as polyunsaturated and saturated fat, cholesterol, and alcohol. The user can select which database to use when starting up the program.



Current status: MacDiet Expanded Database is available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Shortie McKinney, Nutrition and Food Sciences, Drexel University, Philadelphia, PA 19104

Physics Lessons

Programs intended to illuminate particular principles and/or to allow students to investigate phenomena. Completed are Vector, Vector Addition, Simple Pendulum, Fourier, Schrodinger, Logistic Map, and Real Pendulum. The first three are elementary and tutorial, accompanying a first course in physics. Fourier (see figure) illustrates graphically the term-by-term converging approximation of a variety of functions. The last three programs are upper-division level and fully interactive. In Schrodinger, the harmonic oscillator problem is integrated numerically for a specified input value of energy and choice of initial conditions. Divergence for any value other than an energy eigenvalue is graphically demonstrated. Logistic Map exhibits the period doubling approach to chaos: the student specifies values of initial point and growth parameter. Real Pendulum presents graphically the result of a numerical integration of the nonlinear equation that describes the harmonically driven, damped pendulum. All programs are written in Microsoft BASIC, compiled and linked (no license is required for distribution). Copies are available (send disk). At Drake, we distribute the source programs to our upper-division students who have routine access to Microsoft BASIC. They are assigned the development of modifications for further pedagogical investigation. These source codes are available on a reciprocal basis.
Current status: Seven programs completed.
Contact: Dr. Lawrence P. Staunton, Department of Physics and Astronomy, Drake University, Des Moines, IA 50311; (515) 271-3033



Physics

Quantum Mechanics

Quantum Mechanics is designed to implement calculations that are sketched and discussed, but not carried out, in standard quantum mechanics texts adopted at undergraduate and graduate levels. The objectives of these calculations are to determine the transmission and reflection coefficients, or the bound state eigenvalues, of the class of piecewise constant potentials.

Four separate codes calculate and display graphically the transmission coefficient, T(E), as a function of incident particle energy, E, for a scattering potential; the eigenvalues, E, for the eigenstates in a binding potential; the energy bands (both bound and scattering), E(k) as a function of k(_angle), in a periodic potential; and the density of states, r(E), as a function of energy, E, in a periodic potential. Inputs to each code are very similar: the number of pieces in each potential, the energy and width of each, and the energy range to be scanned.

These codes are used to carry out calculations implied, but not done, in standard texts. The results of these calculations are then discussed to illuminate important features not readily apparent in texts. A typical plot of the transmission probability as a function of incident energy is shown. Using this and the code for bound states, the student learns, among other things, that the peaks in the transmission amplitude for a scattering potential determine the locations of the eigenvalues for the corresponding binding potential.

Current status: Quantum Mechanics is available through Kinko's Academic Courseware Exchange (800-235-6919).

Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Interactive Graphics Utilities for Special Relativity

No one can experience directly the world of the very fast that is described by special relativity. Spacetime Software provides interactive graphics displays that help students to visualize this world. In one display, students place clocks, rods, and light flashes on different "speed lanes" of a "cosmic superhighway," jump to different objects on the highway, and ride a "shuttle" that changes lanes. This display is integrated with a conventional space-time diagram that shows events, lightcones, and the "worldliness" of objects moving on the highway. Another display assists in the analysis of relativistic particle collisions, creations, transformations, decays, and annihilations. The operator enters on a table what he or she knows about the incoming and outgoing particles. The machine completes the table, displays the interaction, and plots the energy and momentum of each particle on a perspective three-dimensional graph. Students use these displays as utilities to carry out homework exercises and take-home projects.

Physics

The programs have been developed and tested over three years in special relativity classes at MIT and two years in a continuing education class. The software is supplemented with user manuals, a teacher's guide, sample exercises, and student take-home projects.
Current status: Completed.
Contact: Edwin F. Taylor, Senior Research Scientist, Department of Physics, Room 26-147, Massachusetts Institute of Technology, Cambridge, MA 02139; (617) 253-7433; BITNET: ETAYLOR at MITVMA

Psychology

Linking Cognitive Testing Models to Instruction

Two applications have been developed based on refined digraph analysis and ordering analysis. The first is used in analysis of diagnostic features of an item. The second is used in the analysis of the hierarchical structure among a set of items. The software is written in Pascal.
Current status: Under development.
Contact: Dr. William M. Bart, Educational Psychology, 330 Burton Hall, University of Minnesota, Minneapolis, MN 55455

MacLaboratory for Psychology

MacLaboratory for Psychology provides research-grade software and university-level interactional teaching materials. Most applications have been designed to create a flexible piece of psychological apparatus, so that any number of unique experiments can be conducted; thus, this courseware functions not as a tutor, but rather as a tool for real research activities. Files saved in tab-delimited format can be ported to spreadsheet or database applications for further analysis. The following five-disk set is included in one bundle.

Psychology

Disk 1
Perceptual and Motor Skills: Various experimental settings permit the study of motor skill learning and eye-hand coordination, using such tasks as mirror tracing and pursuit rotor tracking. Data are saved as tab-delimited text files.

Neuropsychology of CNS: This interactive tutorial illustrates the anatomical and behavioral relationships among the major cortical and subcortical structures of the central nervous system.

Disk 2
Problem Solving: At the level of Introductory Psychology, this application is used to investigate the social interaction of group size with task difficulty. Stimulus files and experimental conditions are readily edited, permitting a number of problem-solving tasks. Data are saved as tab-delimited text files.

Hemispheric Function: This software permits tachistoscopic presentation of stimuli to any portion of either visual field under accurate and flexible experimental control. Standard Macintosh features allow for the creation of files for a wide variety and type of visual stimuli. Data are saved as tab-delimited text files.

Disk 3
Pitch Discrimination: Software for the psychophysical study of pitch. Waveform, frequency, amplitude, and duration can be modified and arranged into various sets for auditory testing. Data are saved as tab-delimited text files.

Tone Generator: This supporting application provides an easily edited system to create an infinite variety of waveforms for use in auditory perception experiments. Waveforms can be saved and used in the Pitch Discrimination program.

Disk 4
Survey Questionnaires & Survey Data Summary: These applications facilitate the collection and simple summary of survey data. Both open comments and multiple-choice responses are possible. Survey Questionnaires can output standard text files or files to be analyzed by Survey Data Summary. Survey Data Summary saves its output as a tab-delimited text file.

Disk 5
Neural Modeling in Vision: Spreadsheet templates with accompanying text and figures designed to allow interactive student discovery of some features of retinal circuitry.

Anatomy: Authoring software that allows the instructor to select various regions on diagrams or pictures that can be easily installed to provide instruction in anatomy and physiology. Anatomizer is public domain software used with permission.

Megaroids: This entertaining game is used in the laboratory as an initial "ice-breaker" and source of data for instruction in experimental design. Megaroids is public domain software used with permission.

Published materials for the software listed above include MacLaboratory for Psychology Laboratory Manual, MacLaboratory for Psychology Instructor's Notes, and MacLaboratory for Psychology Courseware Documentation, which can be obtained from McGraw-Hill, College Division, 1221 Avenue of the Americas, New York, NY 10020; (212) 512-2878.

MacLaboratory for Psychology courseware and documentation are available through Kinko's Academic Courseware Exchange, 4141 State St., Santa Barbara, CA 93110; 800-235-6919.

Substantial discounts may be obtained under site-licensing agreements by contacting Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104; (215) 895-2667.

Polygraph: A simulated GSR (Galvanic Silicon Response) that—with appropriate user interaction—can illustrate the orienting response, habituation, classical conditioning, extinction, spontaneous recovery, partial reinforcement, and other concepts from the psychology of learning. Data on GSR latency, amplitude, and duration are saved in tab-delimited text files.

Reaction Time: Completely editable verbal and pictographic stimulus sets can be prepared for presentation in various portions of the visual field under precise experimental control. Traditional reaction-time paradigms are measured to millisecond accuracy. Data are saved as tab-delimited text files.

Psychology

MacLaboratory for Psychology: Part 3

MacLaboratory Controller: The MacLaboratory Controller creates experiment files establishing logical relationships among inputs and outputs. A palette of Boolean operators, clocks, timers, and counters in the software control the hardware interface, which comes in a variety of configurations including A/D and Switch I/O.
Current status: Completed.
Contact: Dr. Douglas Chute, Department of Psychology and Sociology, Drexel University, Philadelphia, PA 19104; (215) 895-1722

Σx_i Statistical Analysis

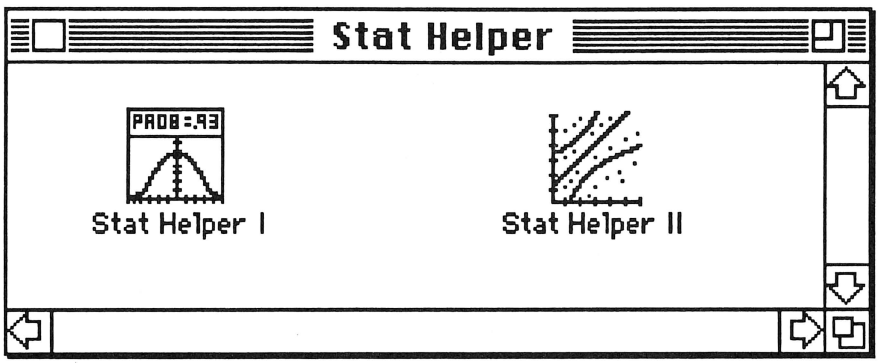
Interactive and Graphical Software for Statistical Education

Software has been developed to teach fundamental statistical ideas. This is not traditional statistical software aimed at calculating statistics and fitting models, rather, the programs demonstrate concepts and teach skills. The nine modules are currently in use in a highly intensive introductory statistics course aimed at graduate students in disciplines other than statistics.
Current status: In use; undergoing testing and evaluation.
Contact: Dr. Kathryn Chaloner, Department of Applied Statistics, 352 Classroom-Office Bldg., University of Minnesota, St. Paul, MN 55108

Stat Helper I and Stat Helper II

Stat Helper I and II are interactive programs that are designed to guide students through the process of statistical problem solving. They are meant to be used in the first and second halves of a two-term undergraduate statistics course. These applications differ from many other statistical software packages in that they do not allow the manipulation of raw data and they won't accept long columns of numbers. (They won't even compute a mean for you!) The goals of these programs are to help students learn to structure the way they read problems, to identify key elements of problems, and to recognize the types of problems that arise in statistics. The basic concept behind these programs is the notion that problem solving need not be a mysterious intuitive process. An experienced statistician approaches problems by asking a sequence of fairly simple questions, and is then able to identify or classify the problem. Once this process is completed, the few bits of information that are hidden in the wording of the problem must be organized. Stat Helper I covers these topics: counting problems, basic probability, discrete probability distributions, continuous probability distributions, sample size calculations, and confidence interval calculations of a mean, a proportion, and a variance. Stat Helper II

covers the following topics: hypothesis testing of means, proportions, and variances; sample size calculations; analysis of variance; and simple regression.



At the heart of these packages is a decision tree, which leads students through the problem identification phase. The tree was designed to give students tools to develop their problem-solving abilities.

There are two versions of the applications: a student version and an instructor version. The student version comes with several word problems, and the instructor version allows the user to create new word problems. Input screens help students organize numerical information that they extract from the problems.

Current status: Stat Helper I and II are available through Kinko's Academic Courseware Exchange (800-235-6919).
Contact: Dr. Janice Biros, Coordinator Microcomputing Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Wheels for the Mind Reader Survey

Dear *Wheels for the Mind* Reader:

We hope that you are enjoying the information about Macintosh use in higher education brought to you each quarter by *Wheels for the Mind.* We are also very interested in your reactions to this publication. To help us ensure your enjoyment of future issues, we would appreciate your cooperation in filling out this survey about the current organization and content of *Wheels for the Mind.* Please write in your answers on the printed tear-out form, and then mail the form to the address given on the "We Want to Hear" form in the front of this issue.

Name, Title, and Institution _____

Street Address _____

City, State, and Zip Code _____

Be sure to answer all questions, bearing in mind that we are interested in your candid opinions.

I. Overall Opinion About *Wheels for the Mind* Sections

1. What information is currently VERY useful to you and your organization or institution?

a. Focus Section: Concentrates on a different theme in each issue. Includes overview, case studies, and additional resources that support the theme.

Continue	Discontinue	Change
[ ]	[ ]	[ ]

Comments_____

b. Spotlight Section: Highlights some of the Macintosh activities going on at various schools and universities.

Continue	Discontinue	Change
[ ]	[ ]	[ ]

Comments_____

c. News and Notes Section: Offers information of general interest, including events, resources, Apple news, and information on new Apple products.

Continue	Discontinue	Change
[ ]	[ ]	[ ]

Comments_____

d. Products Section: Highlights new or unusual Macintosh products that are of interest to the academic community.

Continue	Discontinue	Change
[ ]	[ ]	[ ]

Comments_____

e. Project Reports: Includes projects that are being developed at various colleges and universities.

Continue	Discontinue	Change
[ ]	[ ]	[ ]

Comments_____

II. Possible New Sections

1. What do you think the ultimate *Wheels* should consist of?

	INCLUDE	DON'T INCLUDE	NO OPINION
a. HyperCard Section	[ ]	[ ]	[ ]
b. Interviews with faculty, administrators, and students	[ ]	[ ]	[ ]
c. "How to" articles on applications	[ ]	[ ]	[ ]
d. Technical information	[ ]	[ ]	[ ]
e. Question and answer column	[ ]	[ ]	
f. Surveys on what schools are doing	[ ]	[ ]	[ ]
g. Creative solutions to universal problems—as developed by various schools and universities	[ ]	[ ]	[ ]
h. Programming tutorials	[ ]	[ ]	[ ]
i. Actual program listings	[ ]	[ ]	[ ]
j. Product comparison charts	[ ]	[ ]	[ ]
k. Special topics, such as multimedia, authoring, and expert systems	[ ]	[ ]	[ ]

Comments_____

III. About Yourself

1. What Macintosh systems do you have access to at home?_____

_____

2. What Macintosh systems do you have access to at work?_____

_____

3. What are your primary applications for Macintosh in your work?_____

_____

4. Which software products do you use most often?_____

_____

IV. Future Dreams

1. What other recommendations would you make to the planners, designers, and producers of *Wheels*?

_____

_____

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010
TLX 171-576

M1516/E

Apple Macintosh WHEELS FOR THE MIND 1988 Volume 4 Number 2